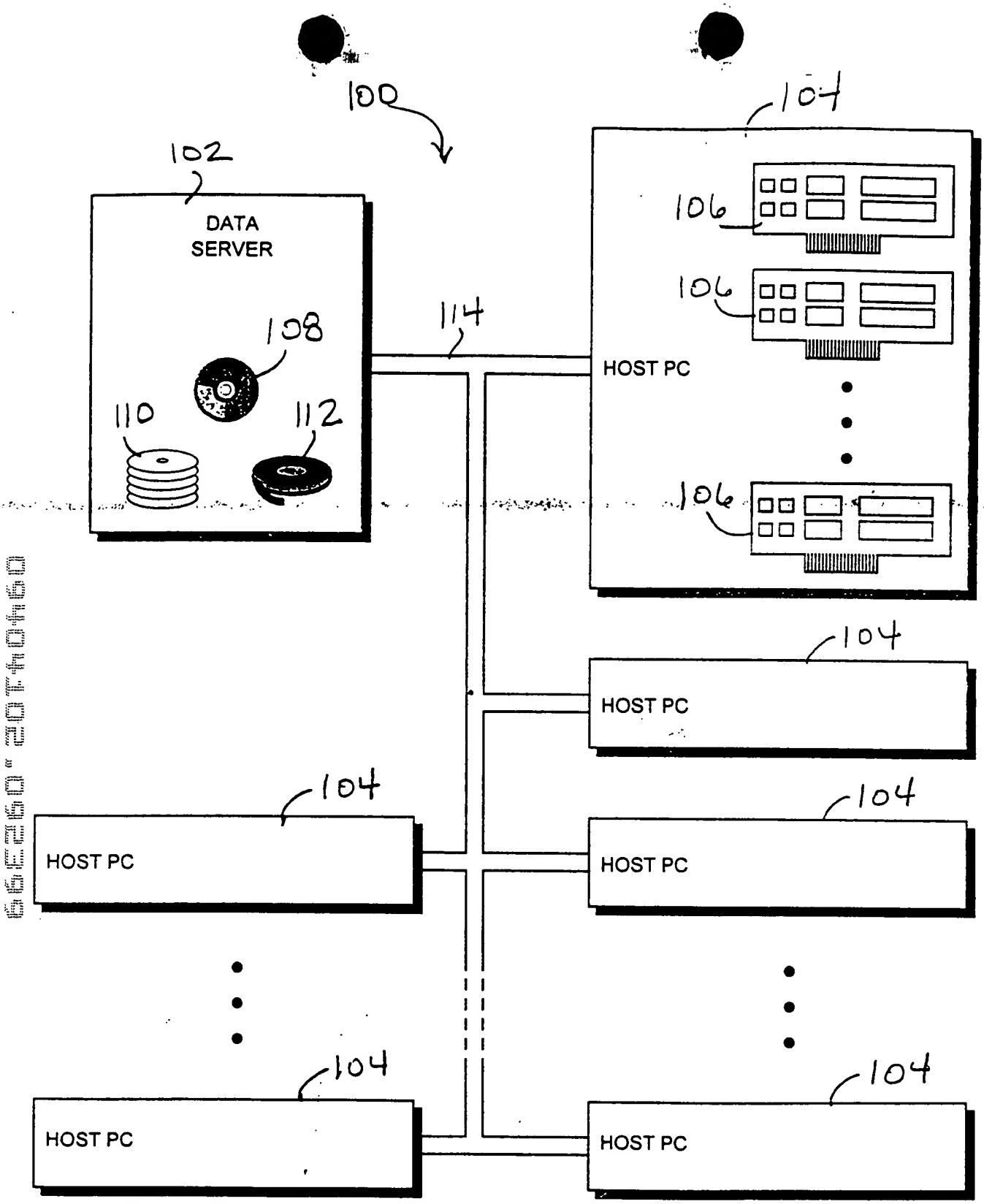

100
102
DATA SERVER
108
110
112
114
104
106
HOST PC
HOST PC
HOST PC
HOST PC
HOST PC
HOST PC
HOST PC
HOST PC

FIG. 1

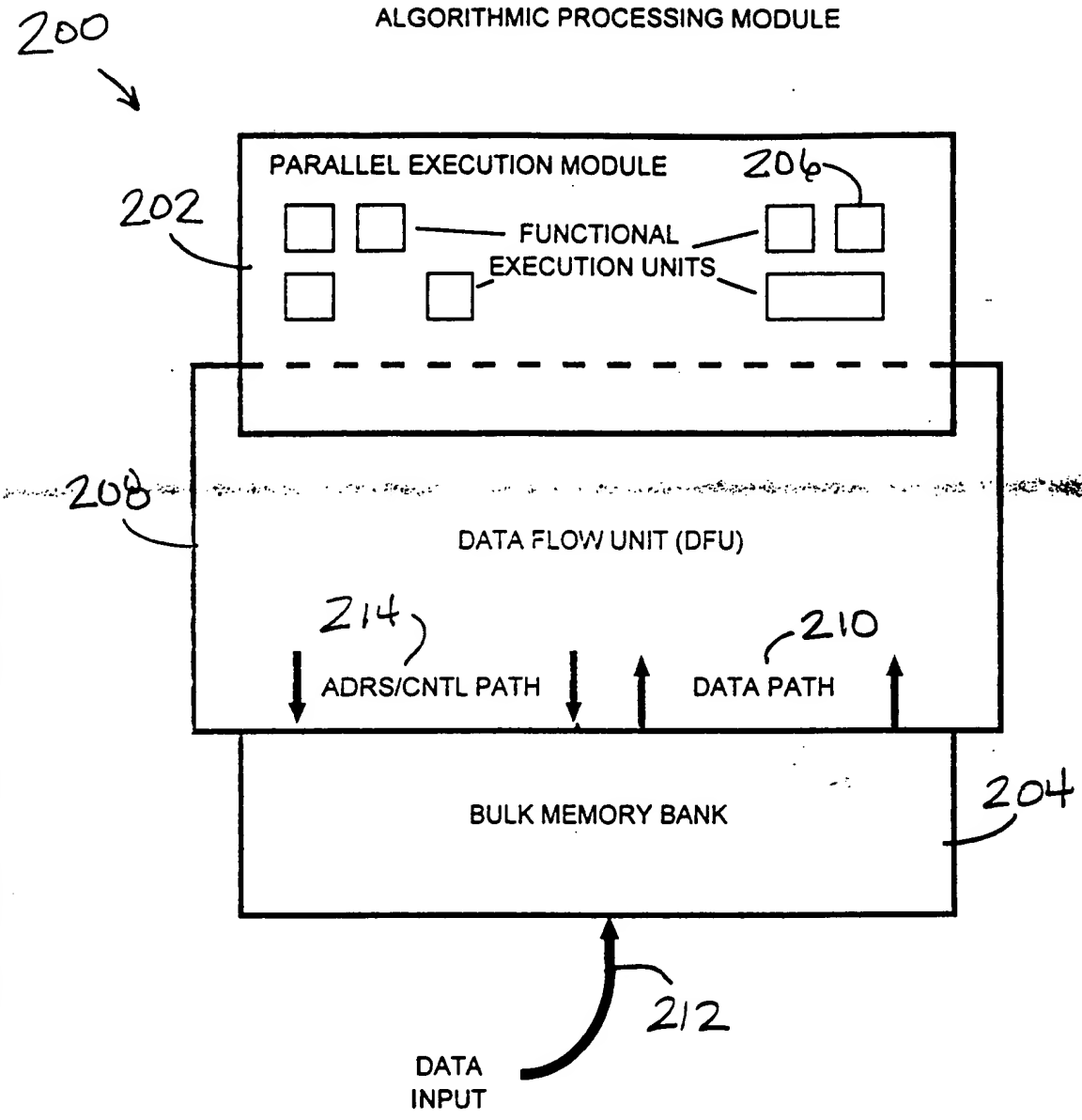

200
ALGORITHMIC PROCESSING MODULE
PARALLEL EXECUTION MODULE
206
202
FUNCTIONAL
EXECUTION UNITS
208
DATA FLOW UNIT (DFU)
214
210
ADRS/CNTL PATH
DATA PATH
BULK MEMORY BANK
204
212
DATA
INPUT

FIG. 2

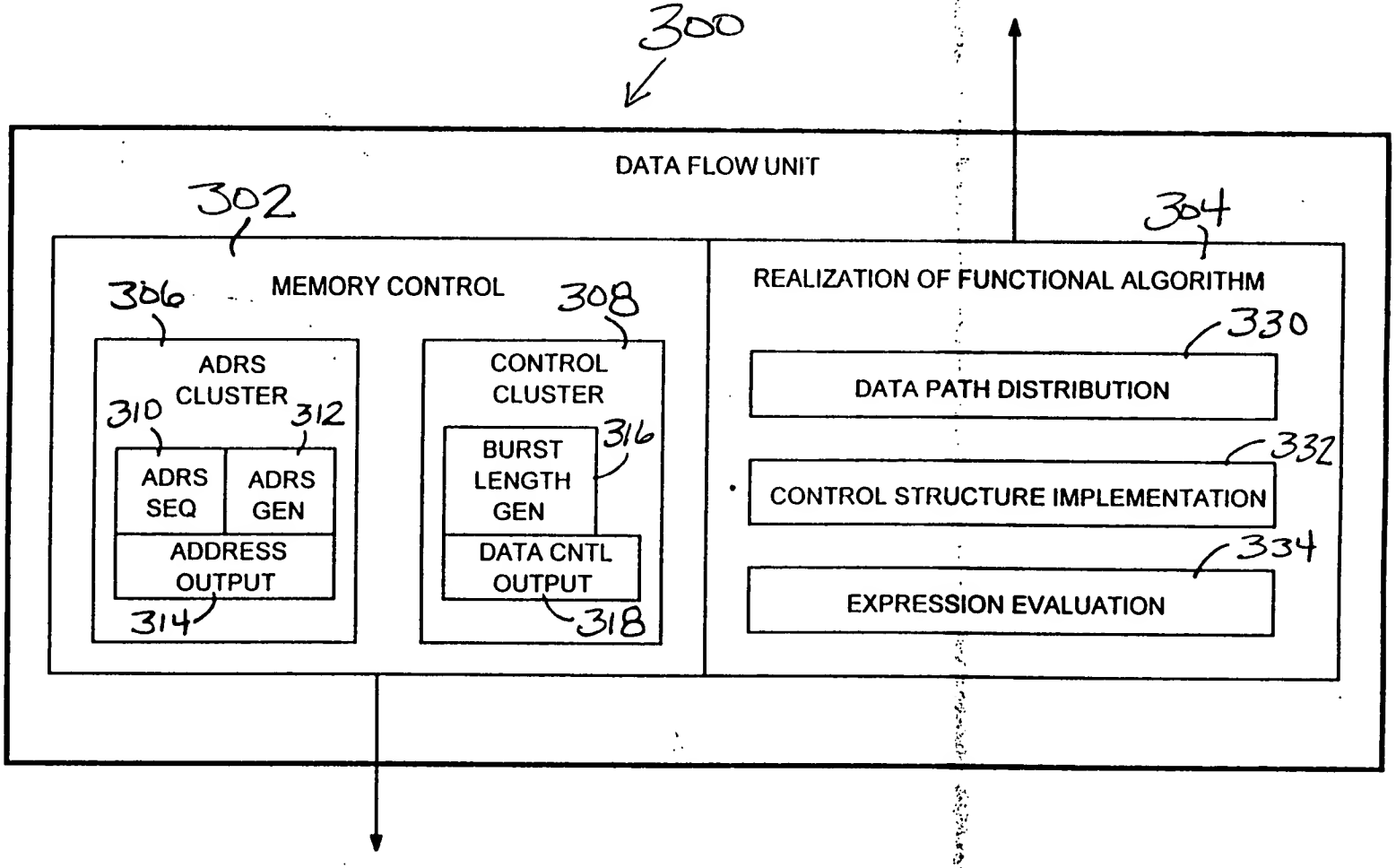


FIG. 3

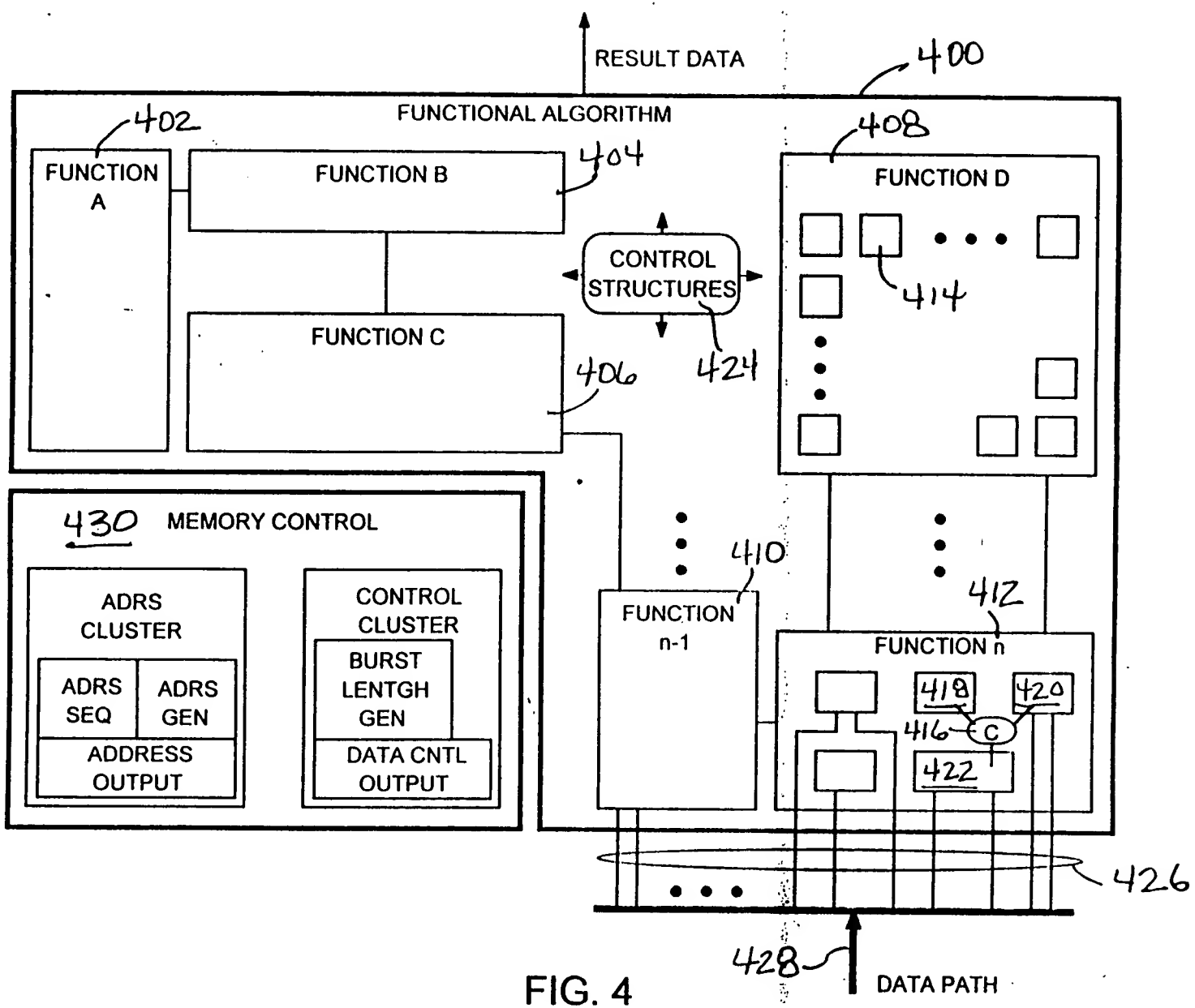


FIG. 4

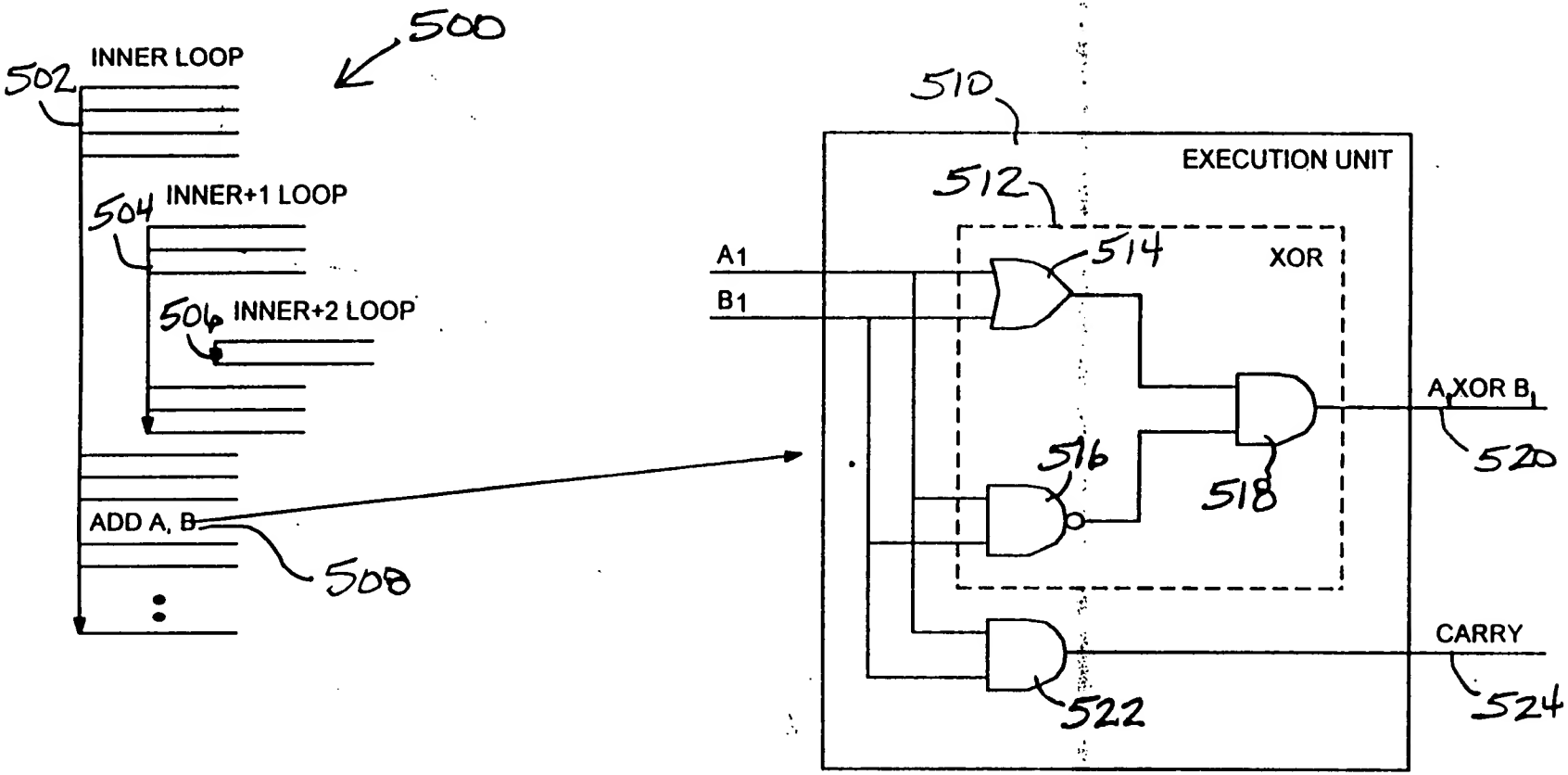

500
INNER LOOP
502
INNER+1 LOOP
504
INNER+2 LOOP
506
ADD A, B
508
510
EXECUTION UNIT
512
XOR
514
A1
B1
516
518
A XOR B
520
CARRY
522
524

FIG. 5

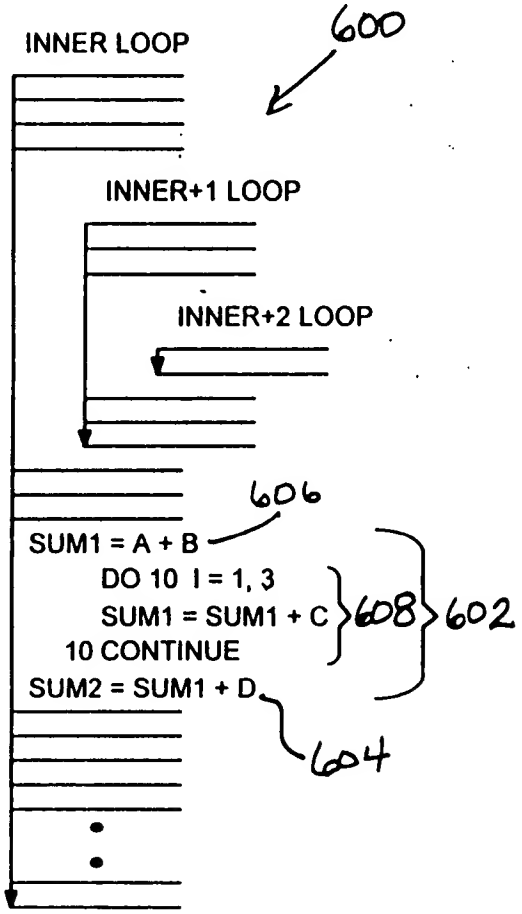


FIG. 6A

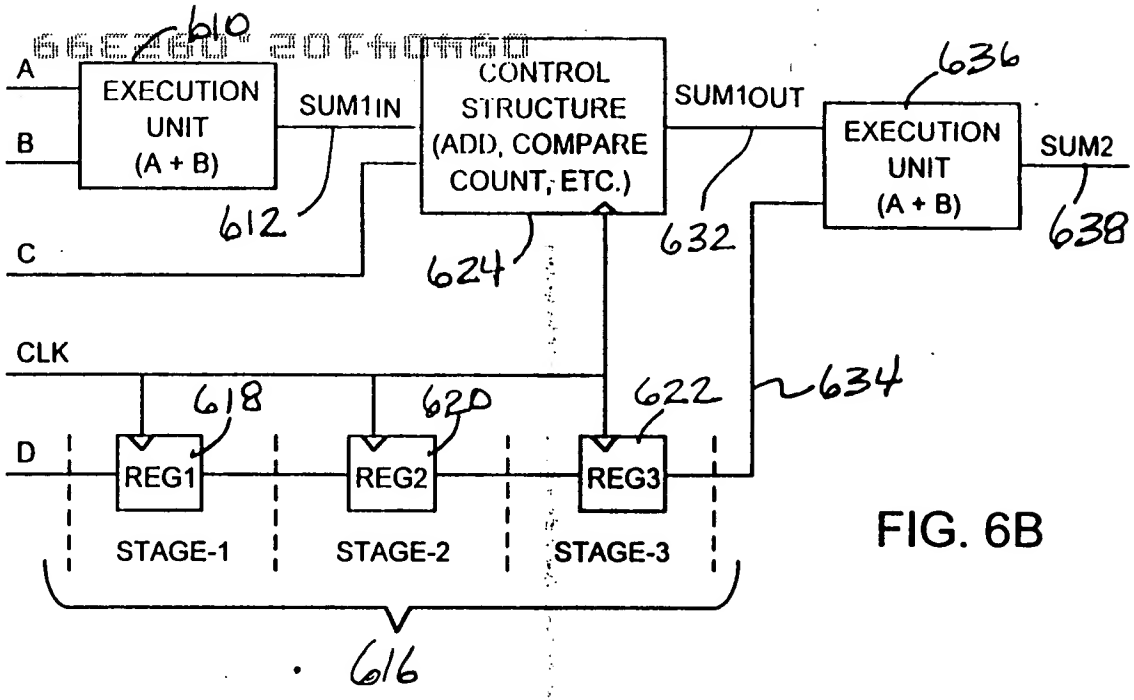


FIG. 6B

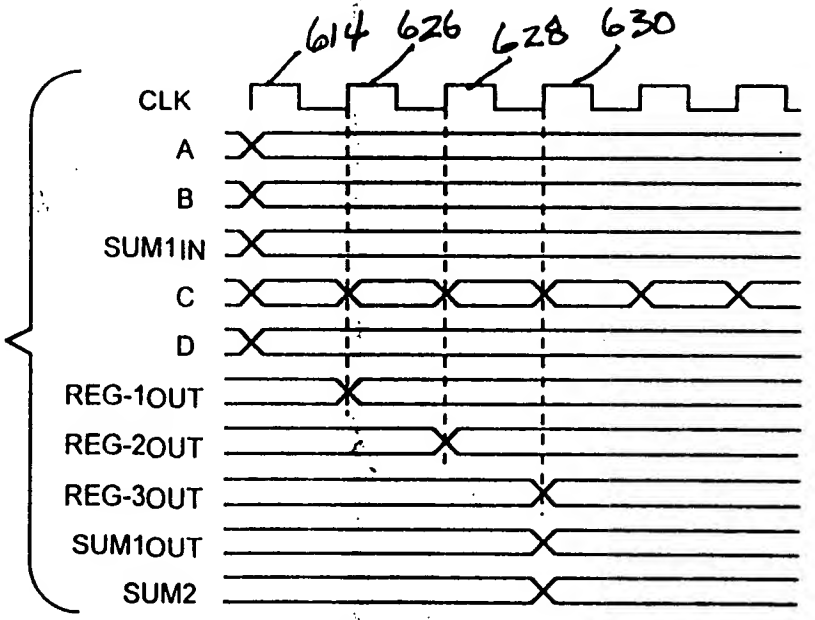


FIG. 6C

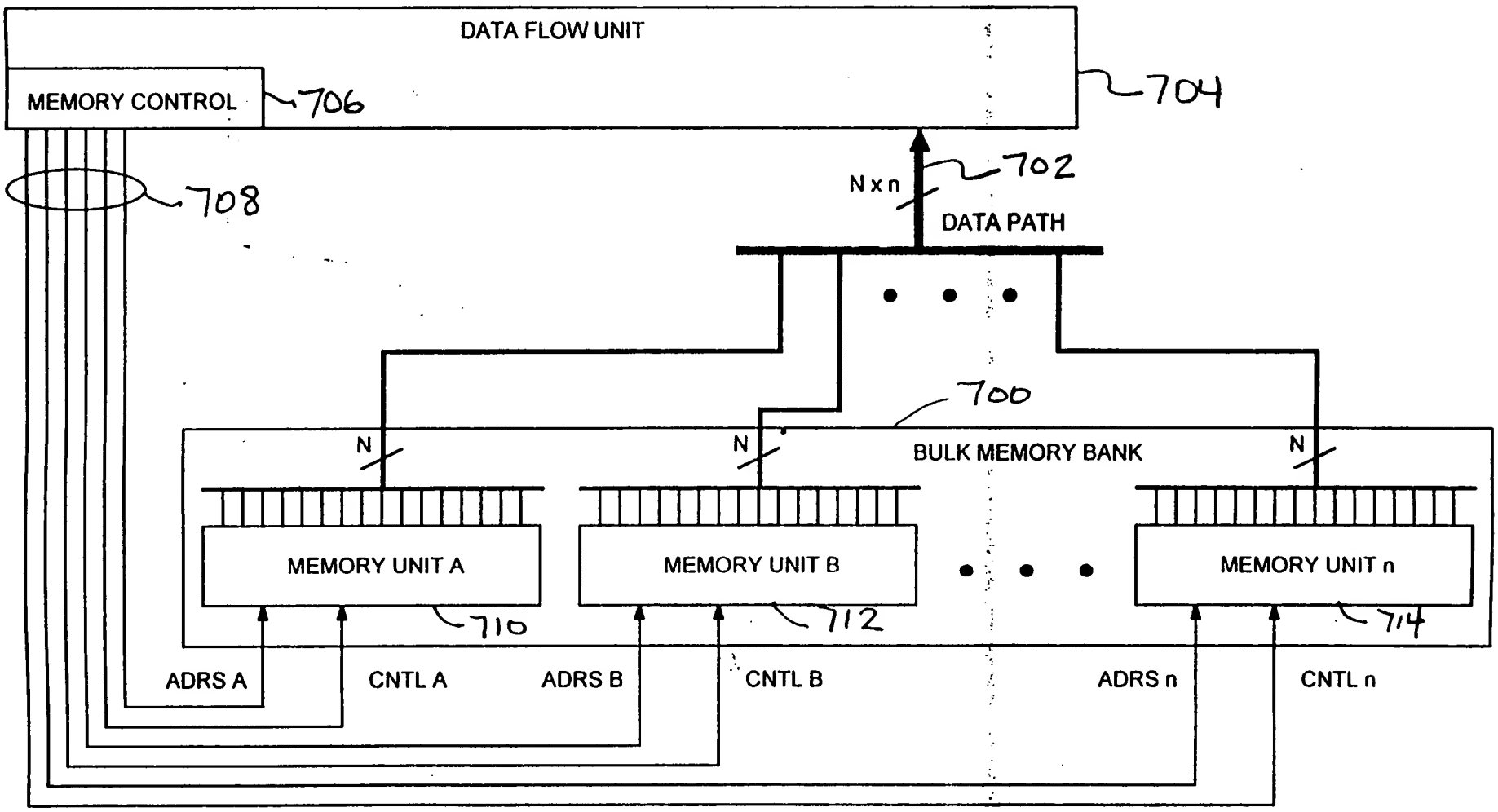

DATA FLOW UNIT
MEMORY CONTROL
706
704
708
N x n
702
DATA PATH
700
N
N
N
BULK MEMORY BANK
MEMORY UNIT A
MEMORY UNIT B
MEMORY UNIT n
710
712
714
ADRS A
CNTL A
ADRS B
CNTL B
ADRS n
CNTL n

FIG. 7

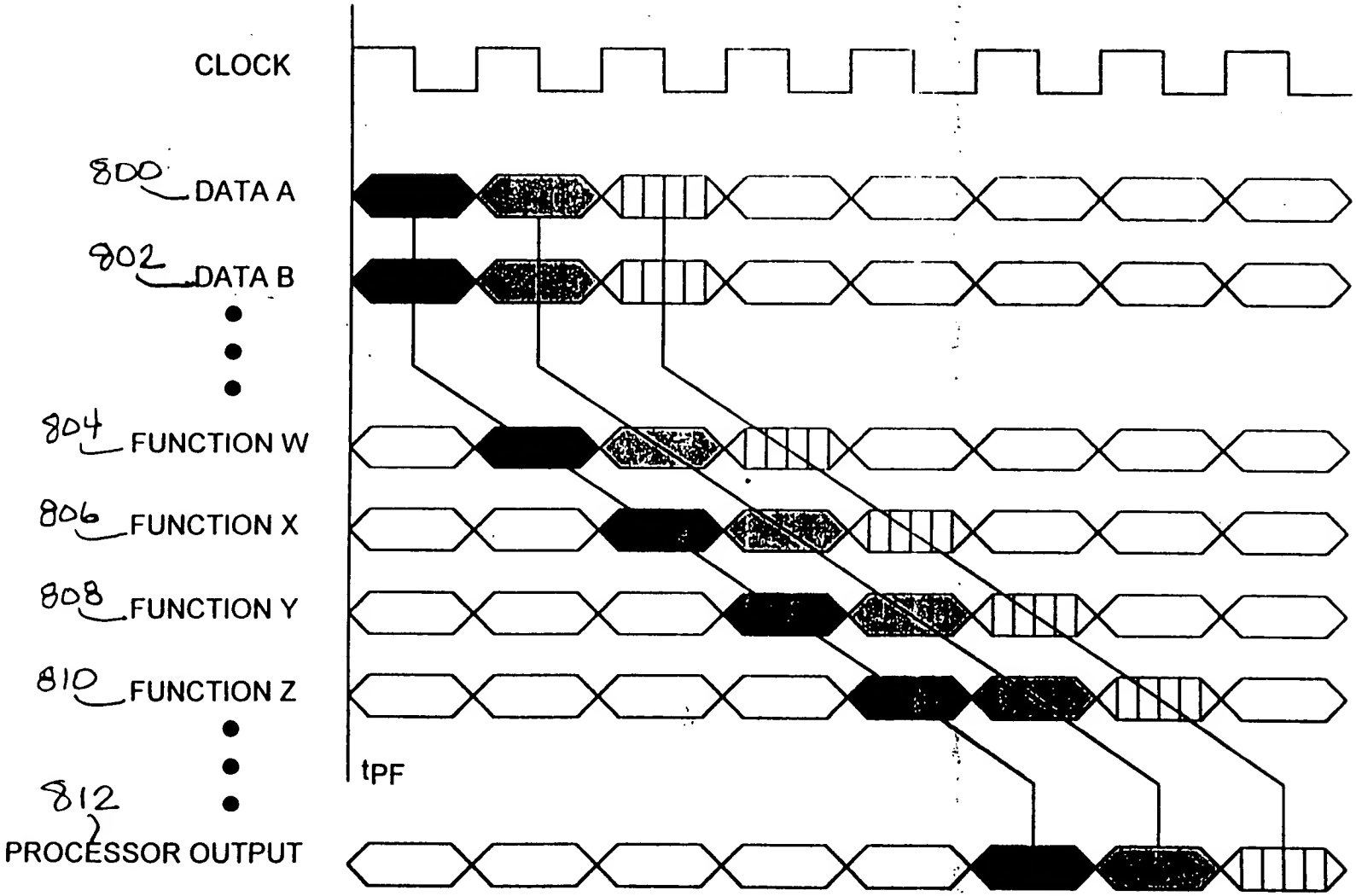

CLOCK
800
DATA A
802
DATA B
804
FUNCTION W
806
FUNCTION X
808
FUNCTION Y
810
FUNCTION Z
812
PROCESSOR OUTPUT
tPF

FIG. 8

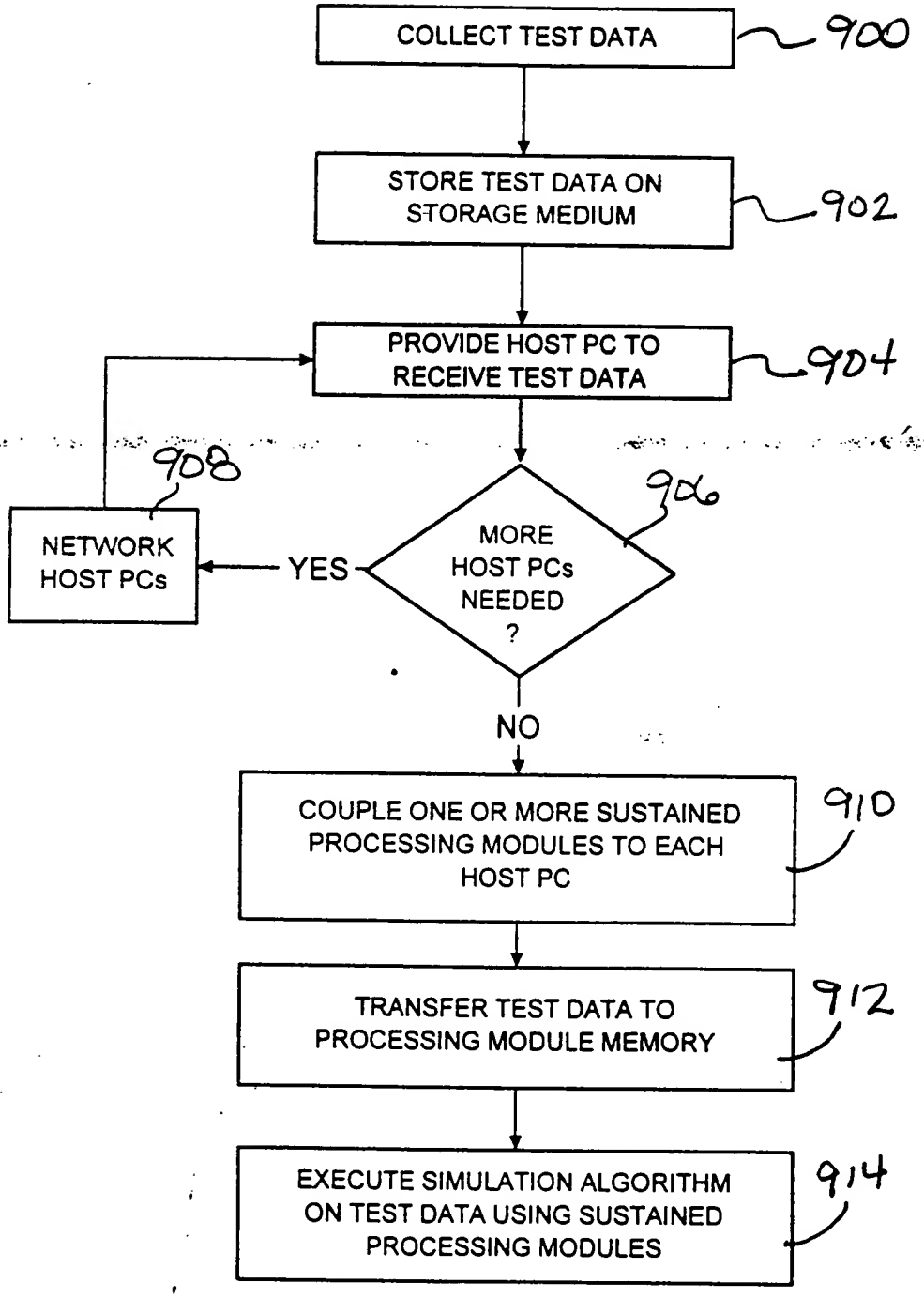

COLLECT TEST DATA
900
STORE TEST DATA ON STORAGE MEDIUM
902
PROVIDE HOST PC TO RECEIVE TEST DATA
904
MORE HOST PCs NEEDED ?
906
YES
NETWORK HOST PCs
908
NO
COUPLE ONE OR MORE SUSTAINED PROCESSING MODULES TO EACH HOST PC
910
TRANSFER TEST DATA TO PROCESSING MODULE MEMORY
912
EXECUTE SIMULATION ALGORITHM ON TEST DATA USING SUSTAINED PROCESSING MODULES
914

FIG. 9

```
      SUBROUTINE AVERAGE(INDATA,COUNT,MAX,MIN,AVG,RMS)
        INTEGER COUNT
        REAL INDATA(COUNT),MAX,MIN,AVG,RMS
C
C These are the inputs and outputs to the function
C  name     type       mode    length     description
C  ====     ====       ====    ======     ===========
C
C  INDATA   R          In      COUNT      Array of input values
C  COUNT    I          In                 Number of values to average
C  MAX      R          InOut              Maximum value found
C  MIN      R          InOut              Minimum value found
C  AVG      R          Out                Average of all values
C  RMS      R          Out                Root-Mean-Square of all values
C
        REAL SUM, RMSSUM
        INTEGER I
C These are internal variables
C Initialization of internal variables
        SUM = 0.0
        RMSSUM = 0.0
C Main Loop proper
        DO 100 I = 1,COUNT
           IF (INDATA(I) .LT. MIN) THEN
             MIN = INDATA(I)
           ENDIF
           IF (INDATA(I) .GT. MAX) THEN
             MAX = INDATA(I)
           ENDIF
        SUM = SUM + INDATA(I)
        RMSSUM = RMSSUM + INDATA(I) * INDATA(I)
  100 CONTINUE
C End of main loop
C Calculate average, RMS
        AVG = SUM / COUNT
        RMSSUM = SQRT(RMSSUM/COUNT)
        END
```

1000 1002 1004 1006 1008 1010 1012 1014 1016 1018

FIG. 10

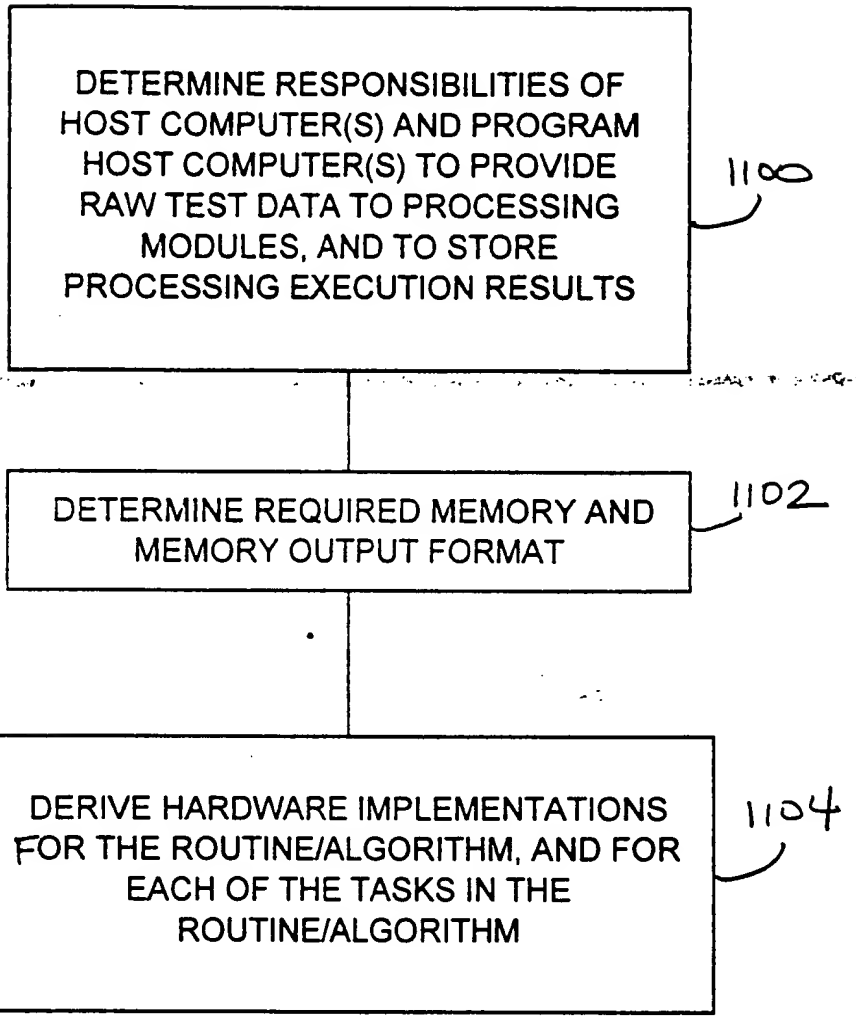

DETERMINE RESPONSIBILITIES OF HOST COMPUTER(S) AND PROGRAM HOST COMPUTER(S) TO PROVIDE RAW TEST DATA TO PROCESSING MODULES, AND TO STORE PROCESSING EXECUTION RESULTS
1100
DETERMINE REQUIRED MEMORY AND MEMORY OUTPUT FORMAT
1102
DERIVE HARDWARE IMPLEMENTATIONS FOR THE ROUTINE/ALGORITHM, AND FOR EACH OF THE TASKS IN THE ROUTINE/ALGORITHM
1104

FIG. 11

```
IF (INDATA(I) .LT. MIN) THEN
      MIN = INDATA(I)
ENDIF
```

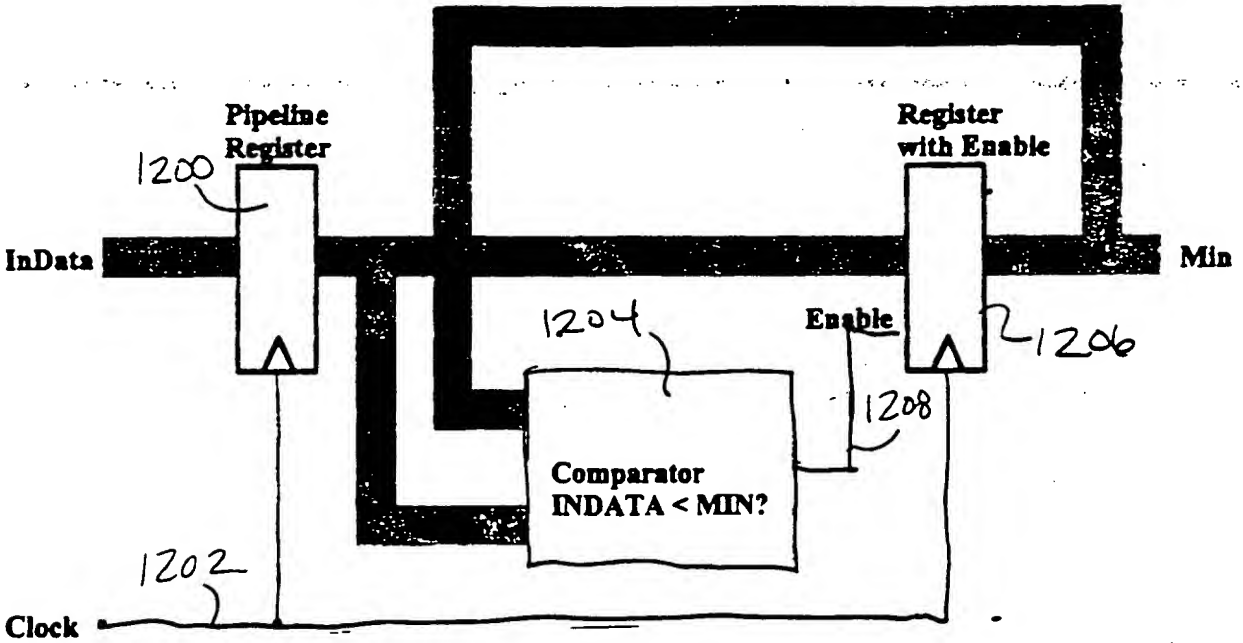


FIG. 12

```
IF (INDATA(I) .LT. MIN) THEN
      MIN = INDATA(I)
ENDIF
```

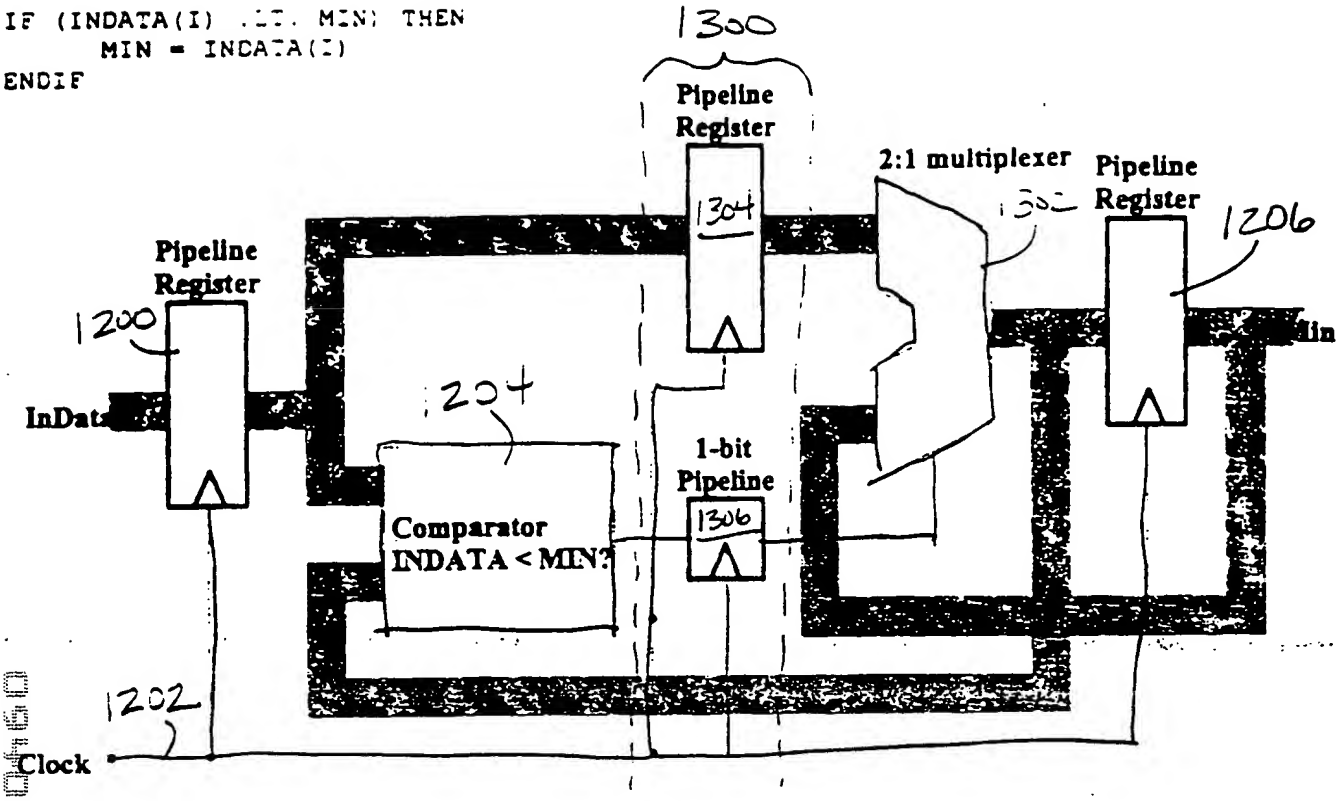


FIG. 13

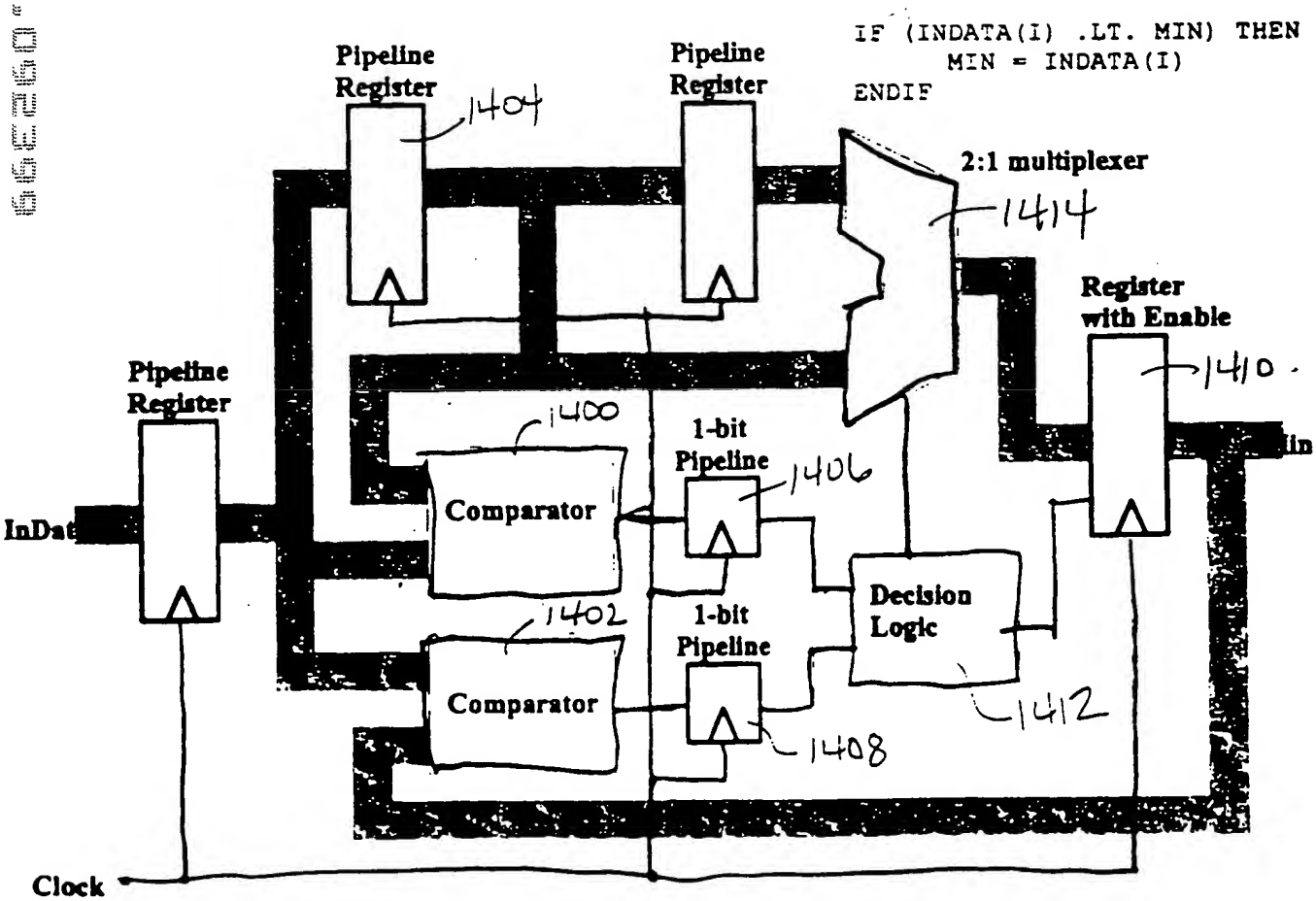


FIG. 14

SUM = SUM + INDATA(I)

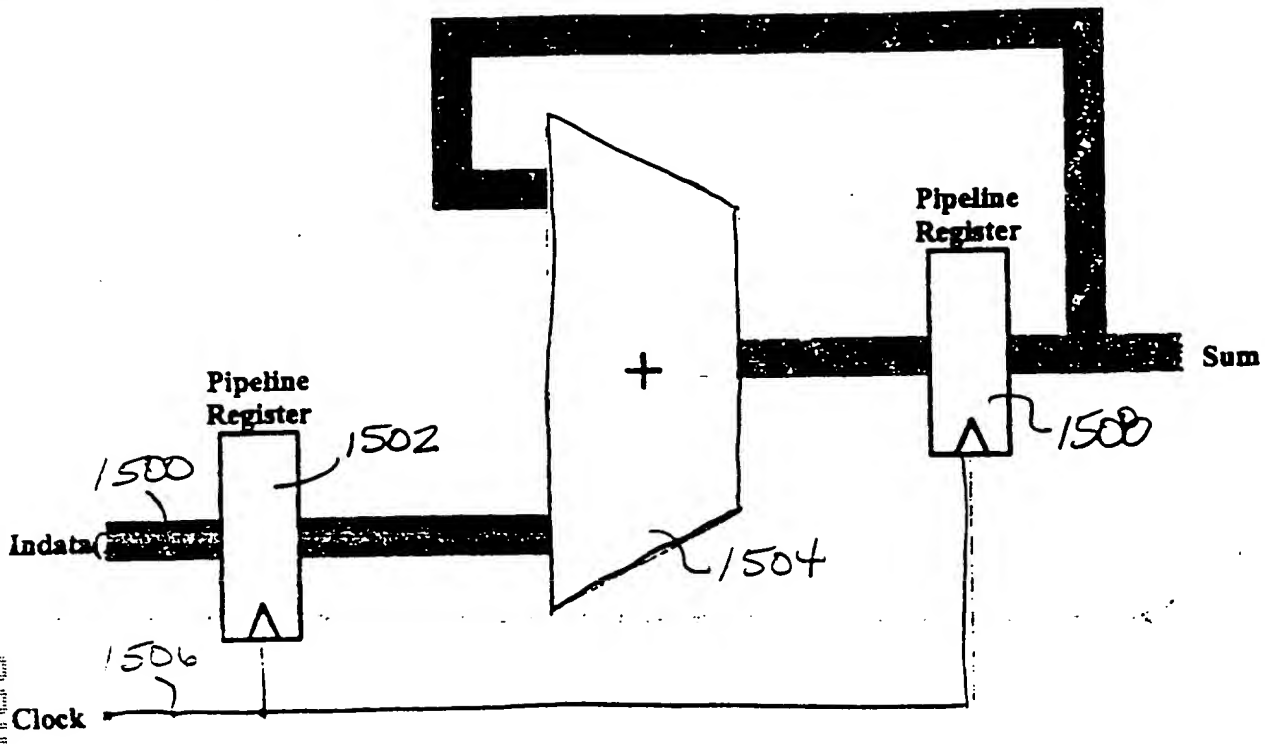


FIG. 15

RMSSUM = RMSSUM + INDATA(I) * INDATA(I)

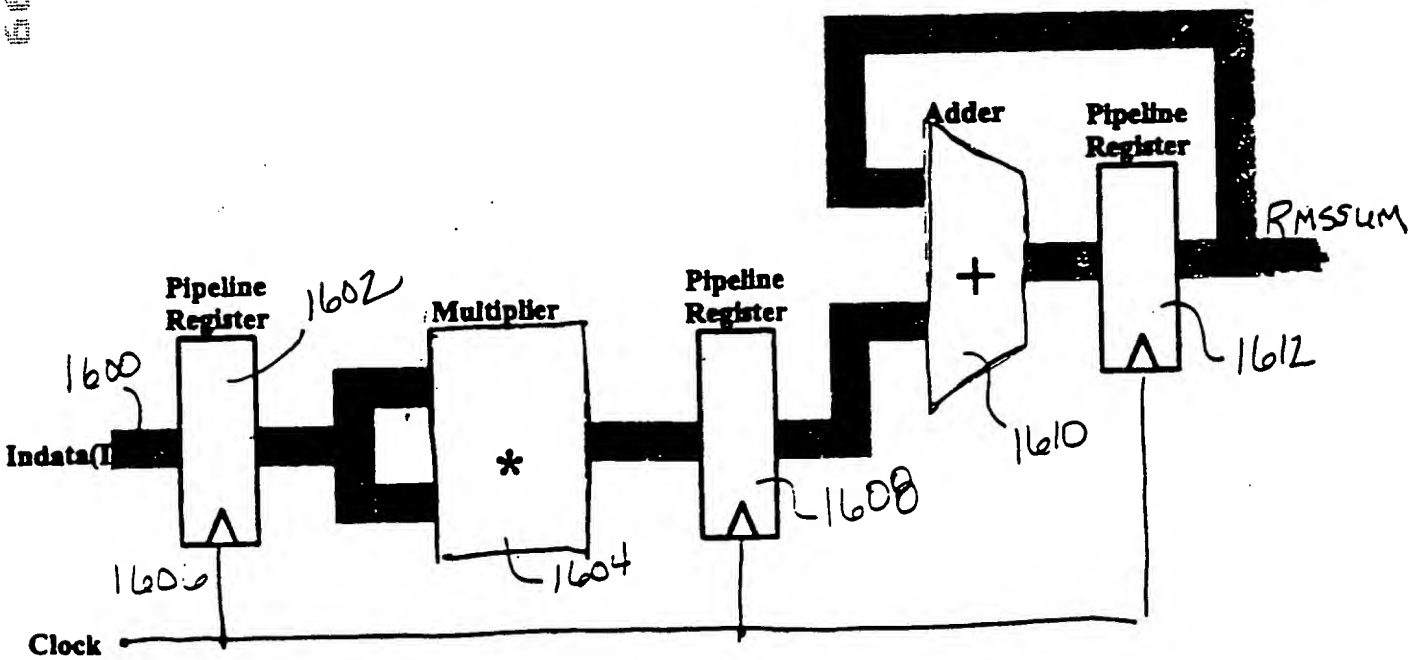


FIG. 16

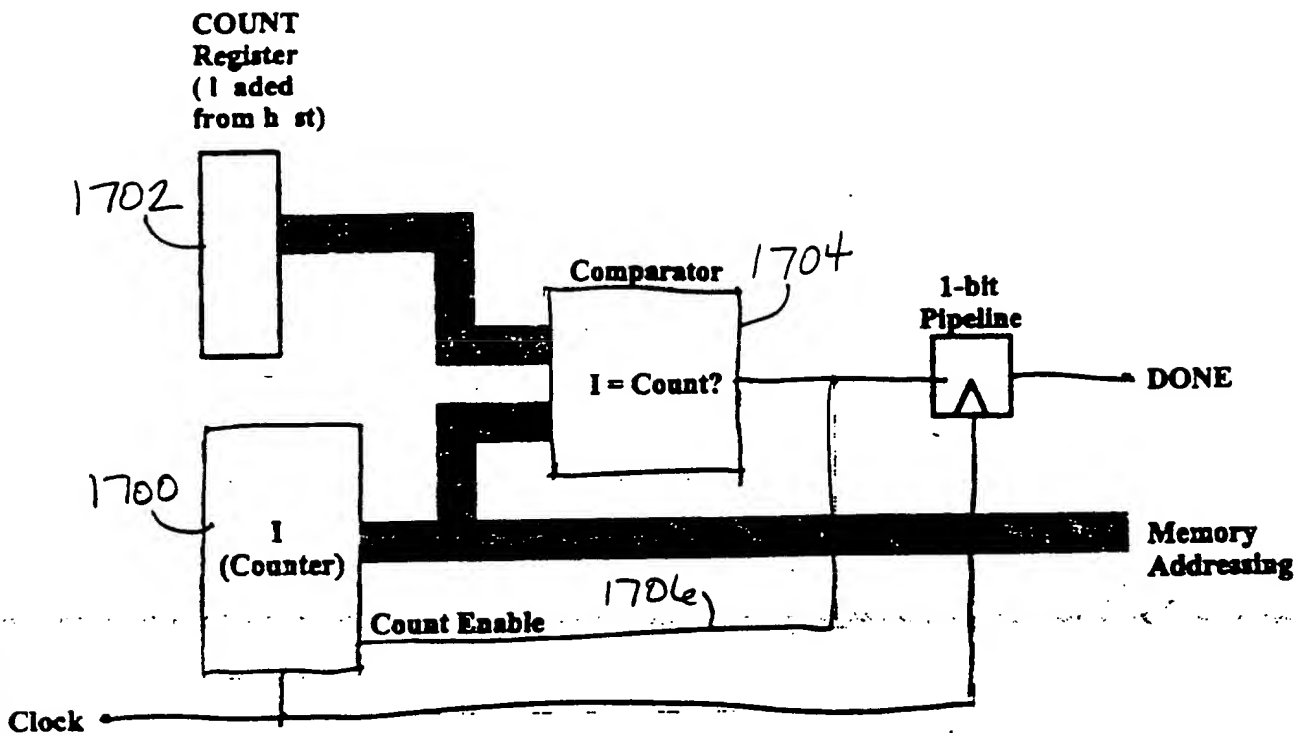

COUNT
Register
(l aded
from h st)
1702
Comparator
1704
1-bit
Pipeline
I = Count?
DONE
1700
I
(Counter)
Memory
Addressing
1706
Count Enable
Clock

FIG. 17

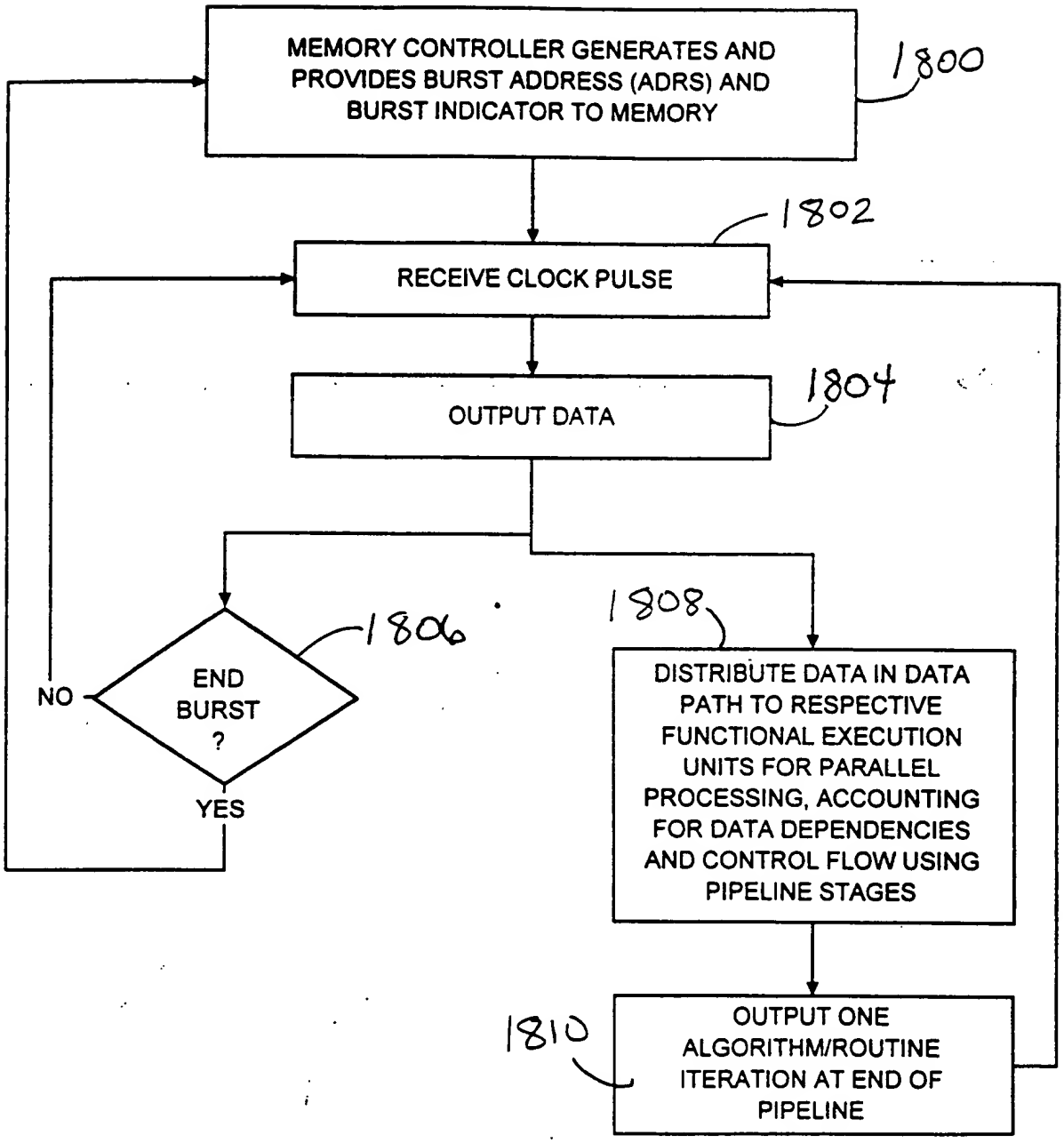

MEMORY CONTROLLER GENERATES AND PROVIDES BURST ADDRESS (ADRS) AND BURST INDICATOR TO MEMORY
1800
1802
RECEIVE CLOCK PULSE
OUTPUT DATA
1804
1806
END BURST ?
NO
YES
1808
DISTRIBUTE DATA IN DATA PATH TO RESPECTIVE FUNCTIONAL EXECUTION UNITS FOR PARALLEL PROCESSING, ACCOUNTING FOR DATA DEPENDENCIES AND CONTROL FLOW USING PIPELINE STAGES
1810
OUTPUT ONE ALGORITHM/ROUTINE ITERATION AT END OF PIPELINE

FIG. 18

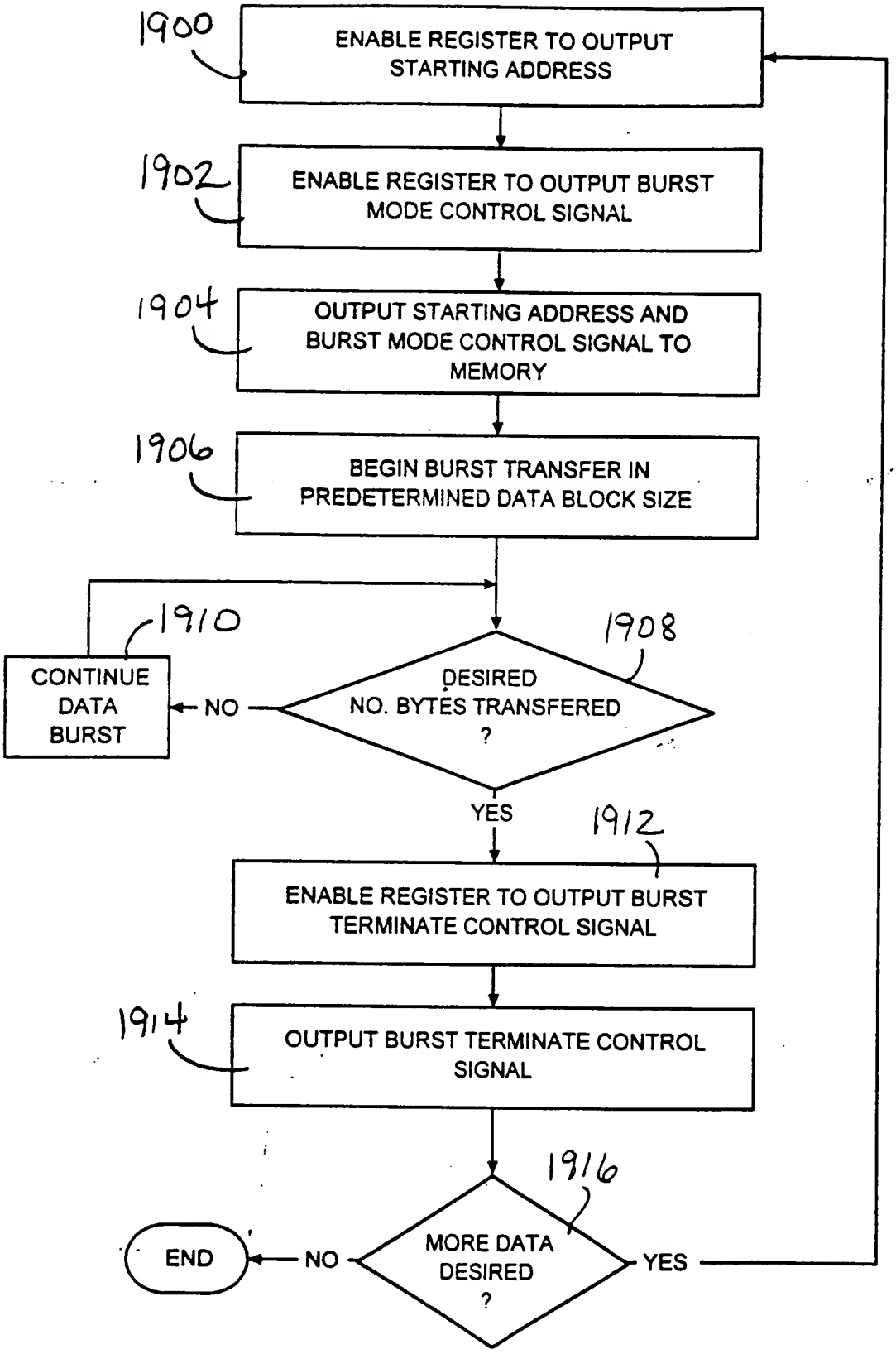

1900
ENABLE REGISTER TO OUTPUT STARTING ADDRESS
1902
ENABLE REGISTER TO OUTPUT BURST MODE CONTROL SIGNAL
1904
OUTPUT STARTING ADDRESS AND BURST MODE CONTROL SIGNAL TO MEMORY
1906
BEGIN BURST TRANSFER IN PREDETERMINED DATA BLOCK SIZE
1908
DESIRED NO. BYTES TRANSFERED ?
NO
1910
CONTINUE DATA BURST
YES
1912
ENABLE REGISTER TO OUTPUT BURST TERMINATE CONTROL SIGNAL
1914
OUTPUT BURST TERMINATE CONTROL SIGNAL
1916
MORE DATA DESIRED ?
NO
END
YES

FIG. 19

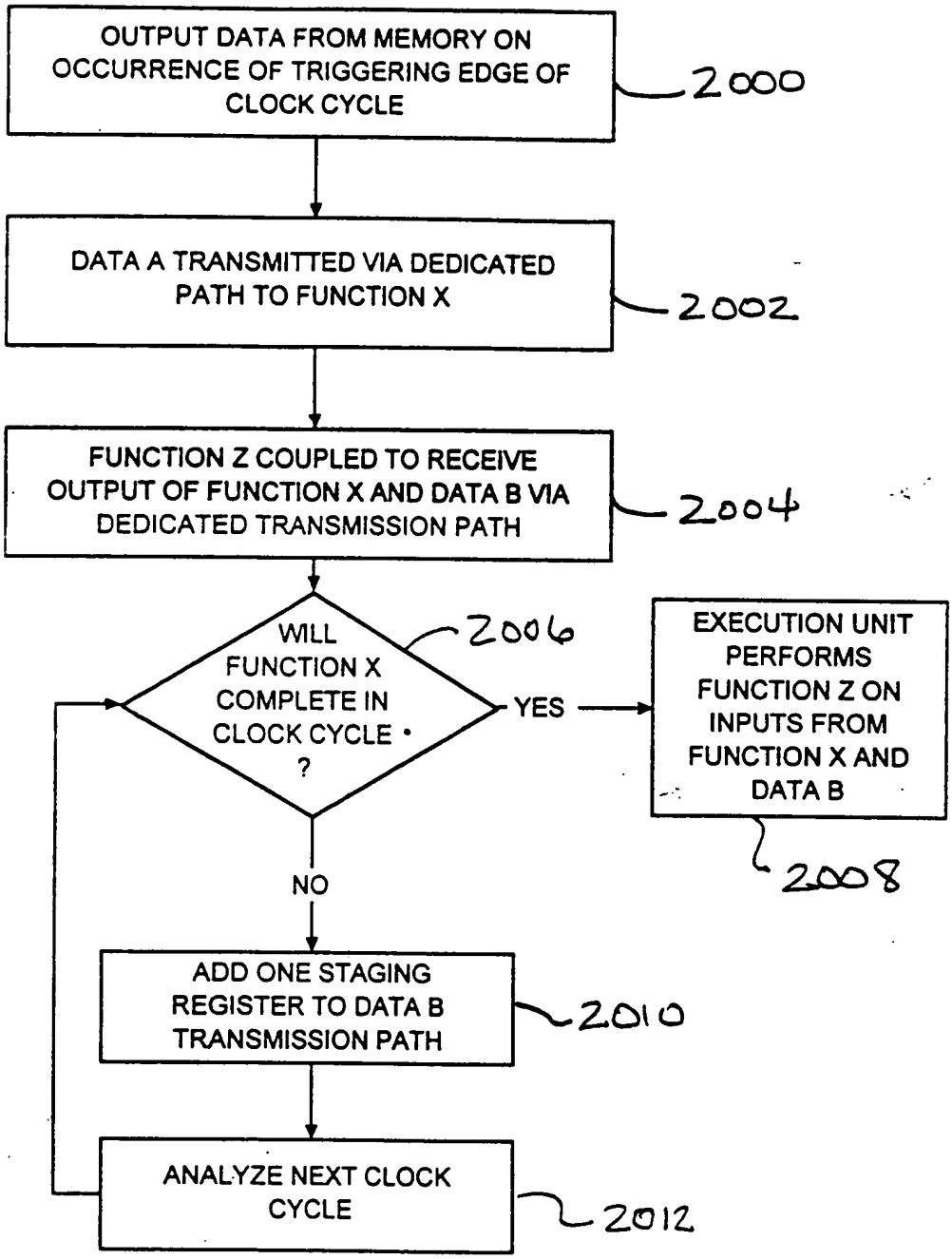

OUTPUT DATA FROM MEMORY ON OCCURRENCE OF TRIGGERING EDGE OF CLOCK CYCLE
2000
DATA A TRANSMITTED VIA DEDICATED PATH TO FUNCTION X
2002
FUNCTION Z COUPLED TO RECEIVE OUTPUT OF FUNCTION X AND DATA B VIA DEDICATED TRANSMISSION PATH
2004
WILL FUNCTION X COMPLETE IN CLOCK CYCLE • ?
2006
YES
EXECUTION UNIT PERFORMS FUNCTION Z ON INPUTS FROM FUNCTION X AND DATA B
2008
NO
ADD ONE STAGING REGISTER TO DATA B TRANSMISSION PATH
2010
ANALYZE NEXT CLOCK CYCLE
2012

FIG. 20

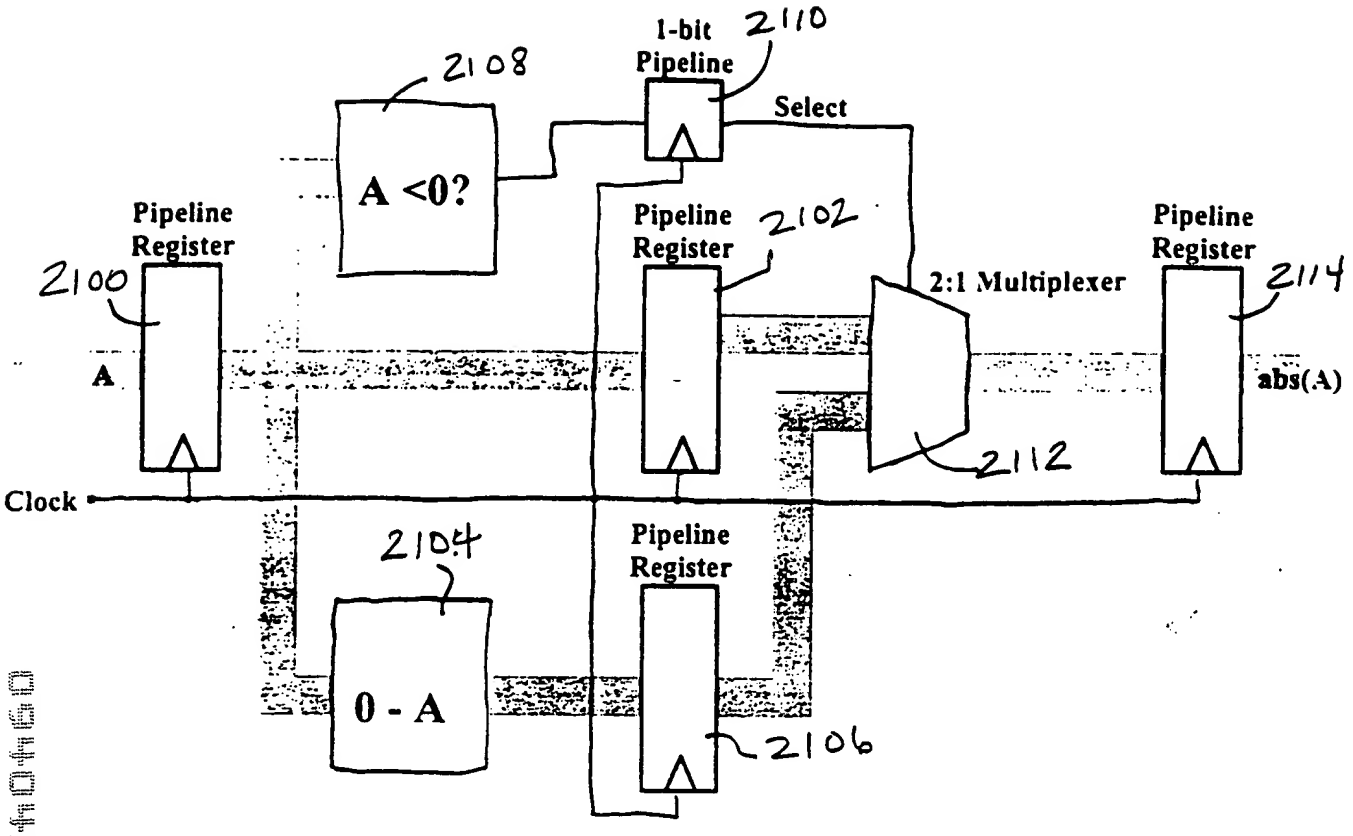


Figure 1. abs(a) naive hardware implementation - 2 cycles

Note:- A width = N bits

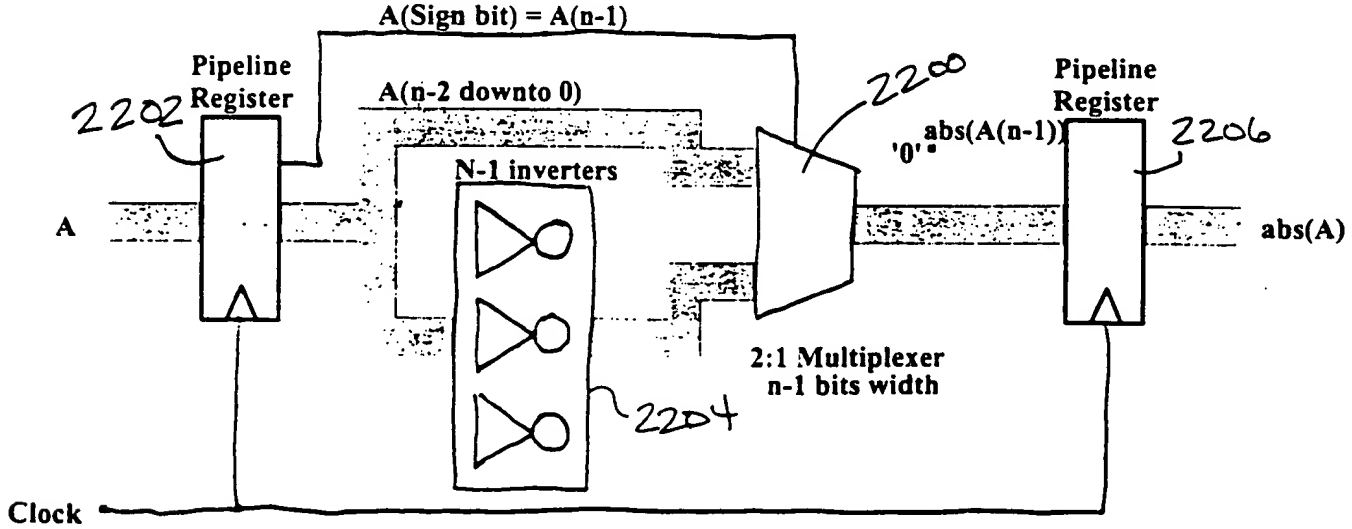


Figure 2. abs(a) faster implementation - 1 cycle

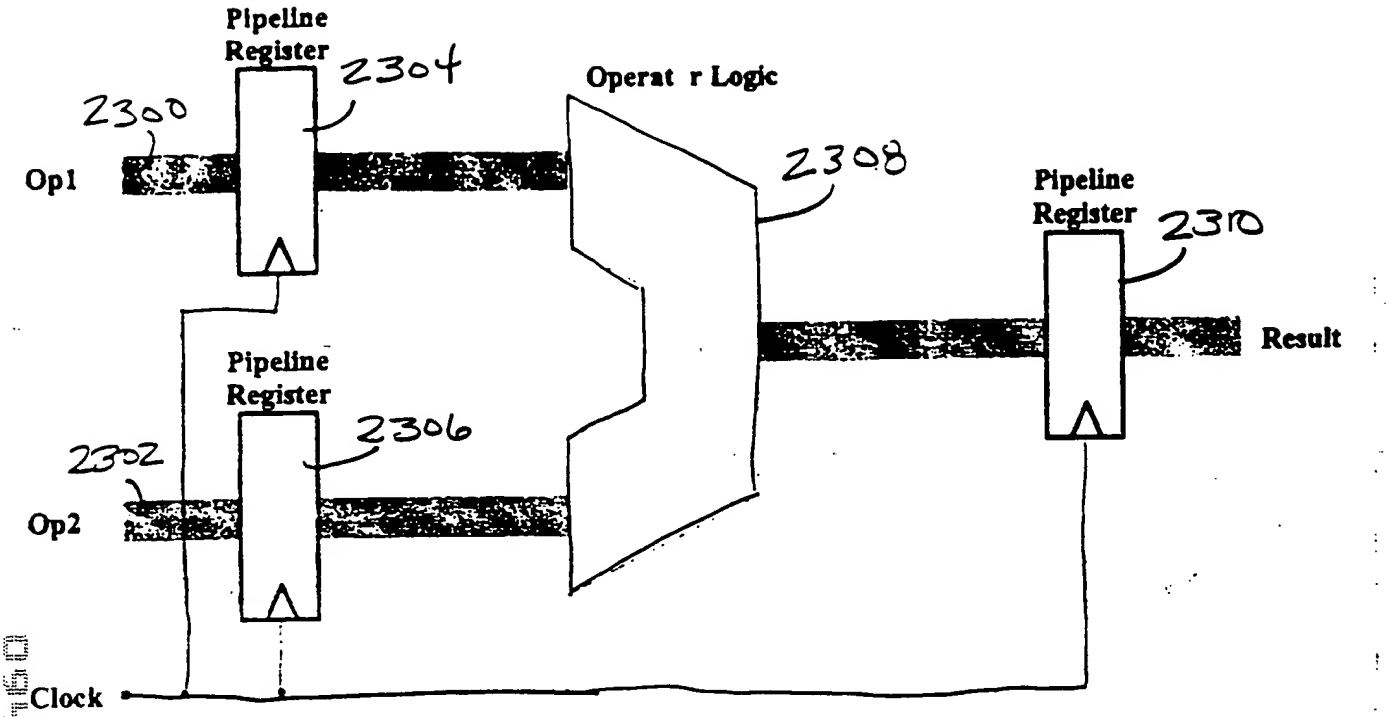


Figure 23 Binary operator

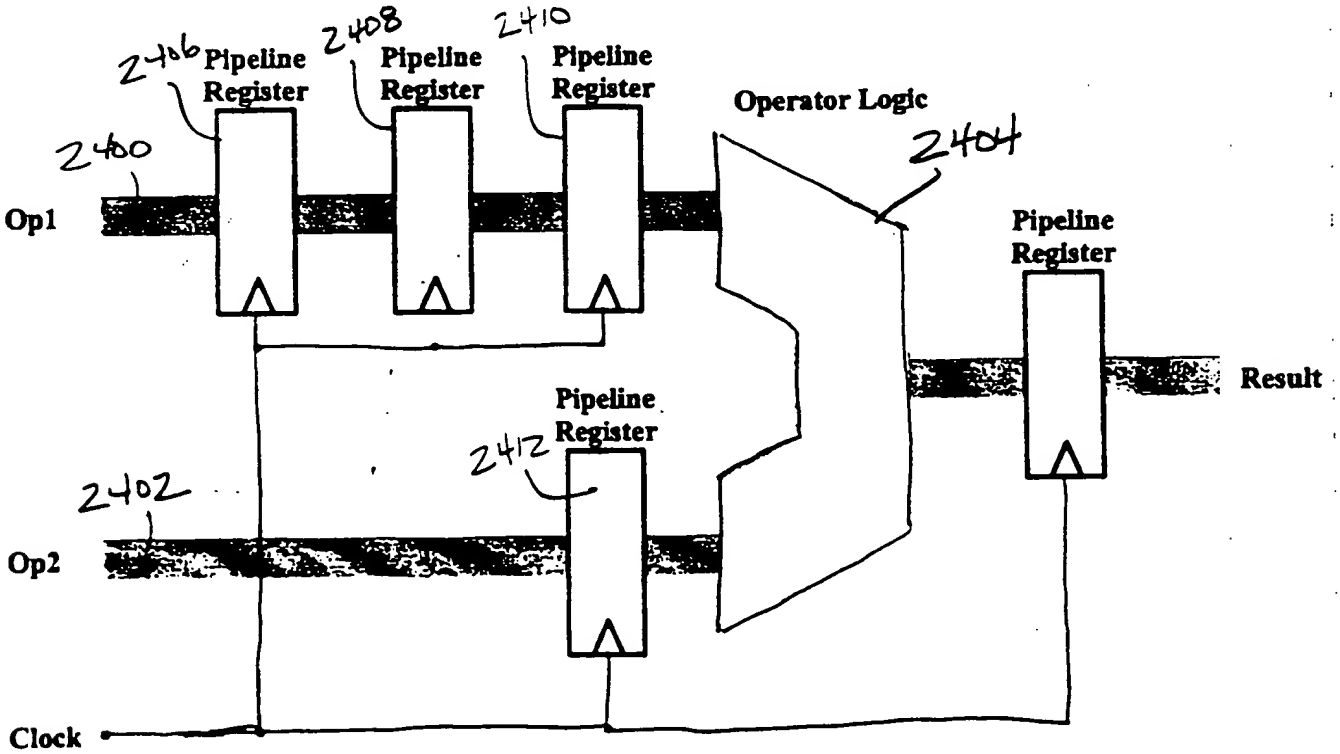


Figure 24 Binary operator with one early operand

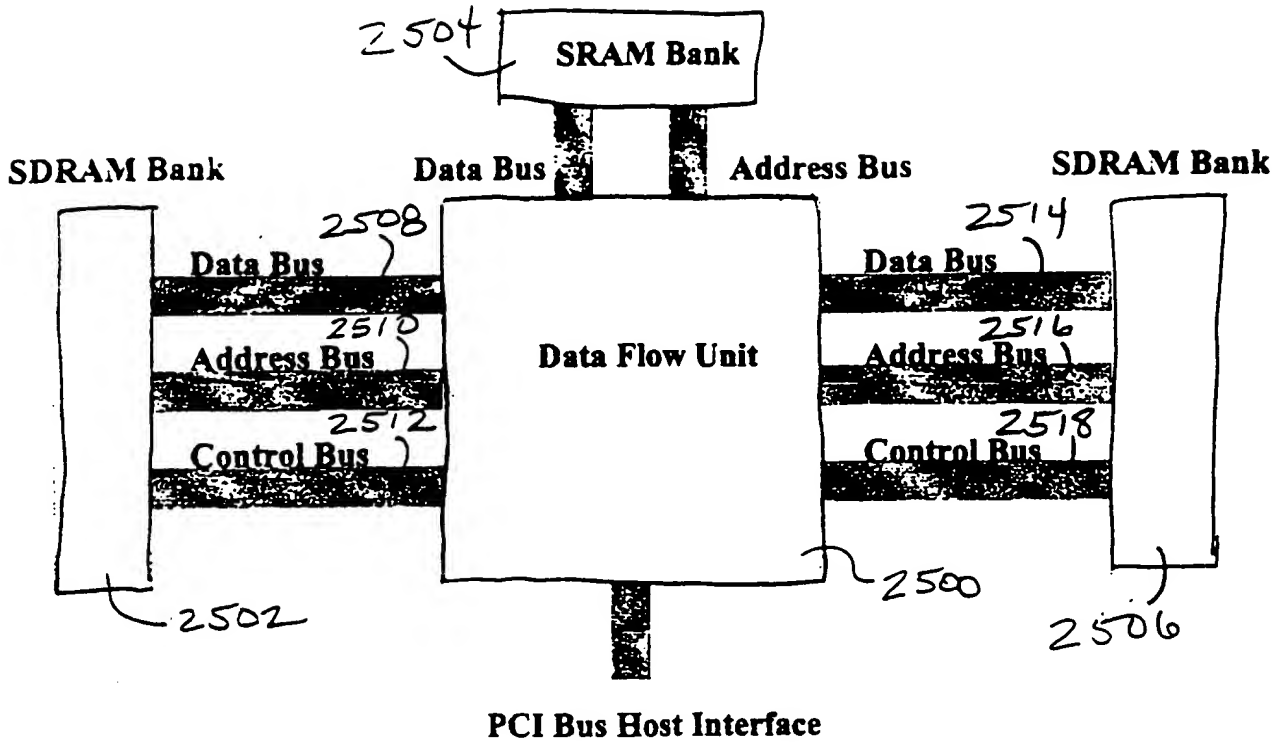


Figure25 Typical system block diagram

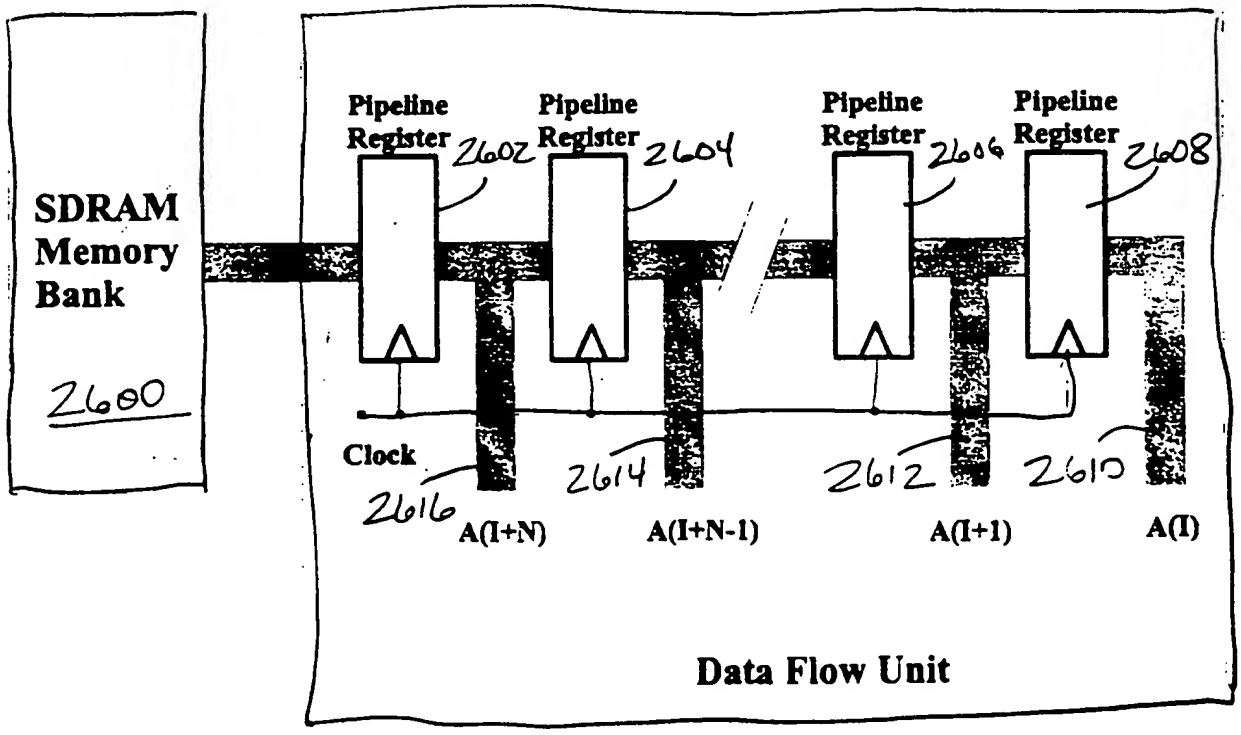


Figure26 Pipelining to reduce memory accesses.

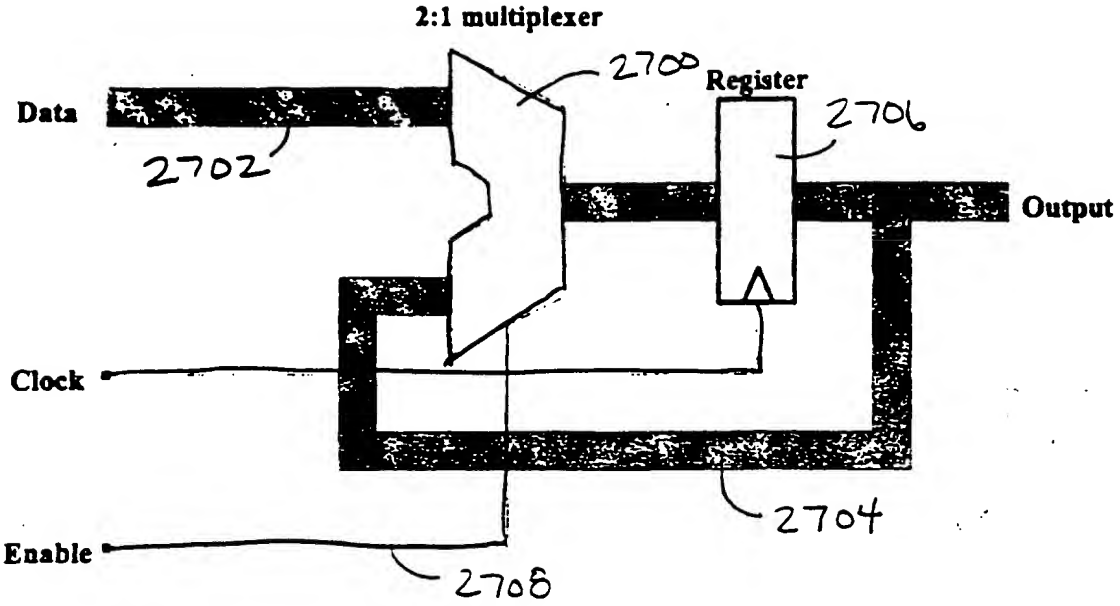


Figure 27 Implementation of Register with Enable

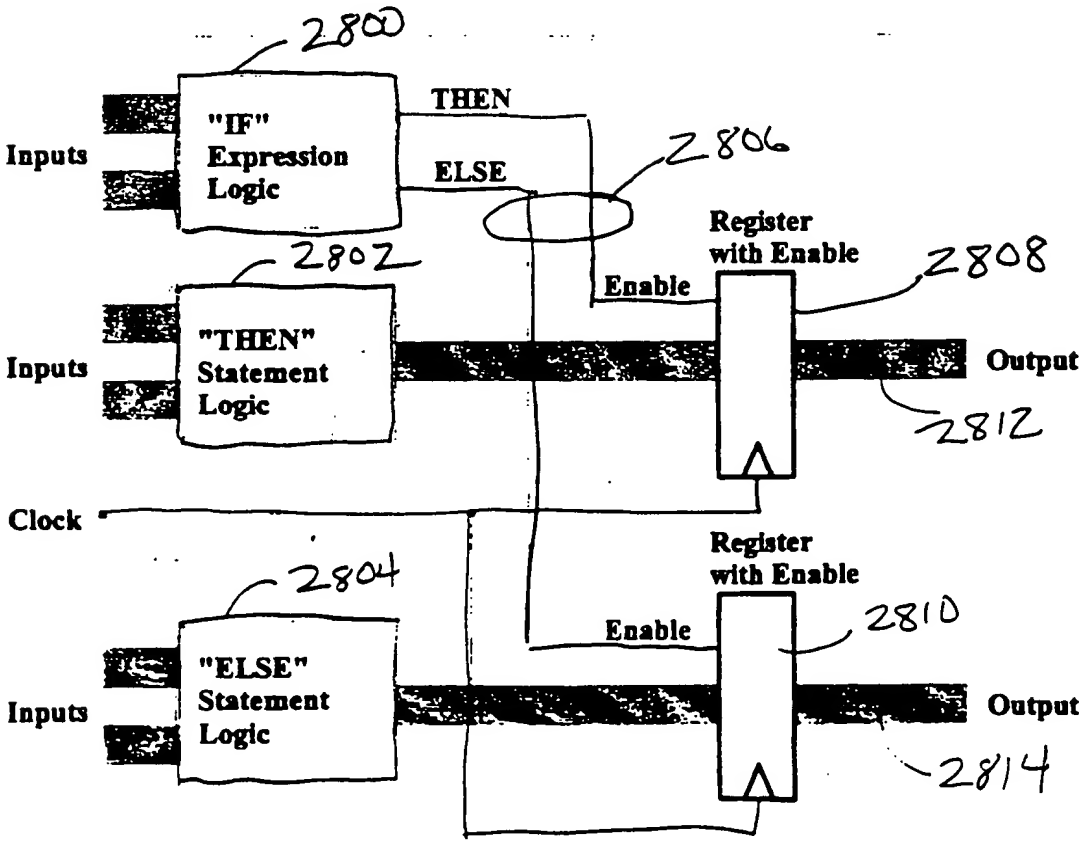


Figure 28 Hardware implementation of conditional statement.

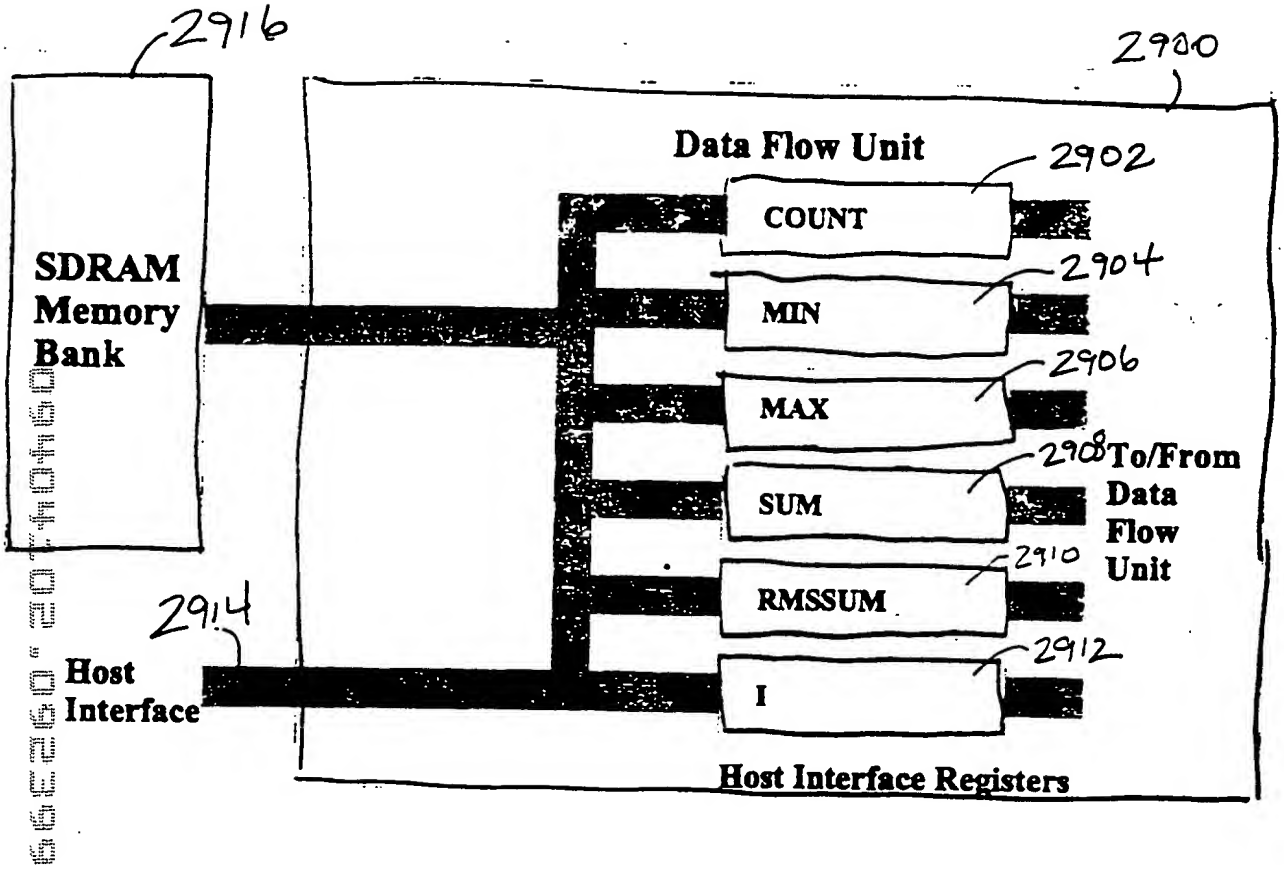


Figure 29 Example host interface/programming model

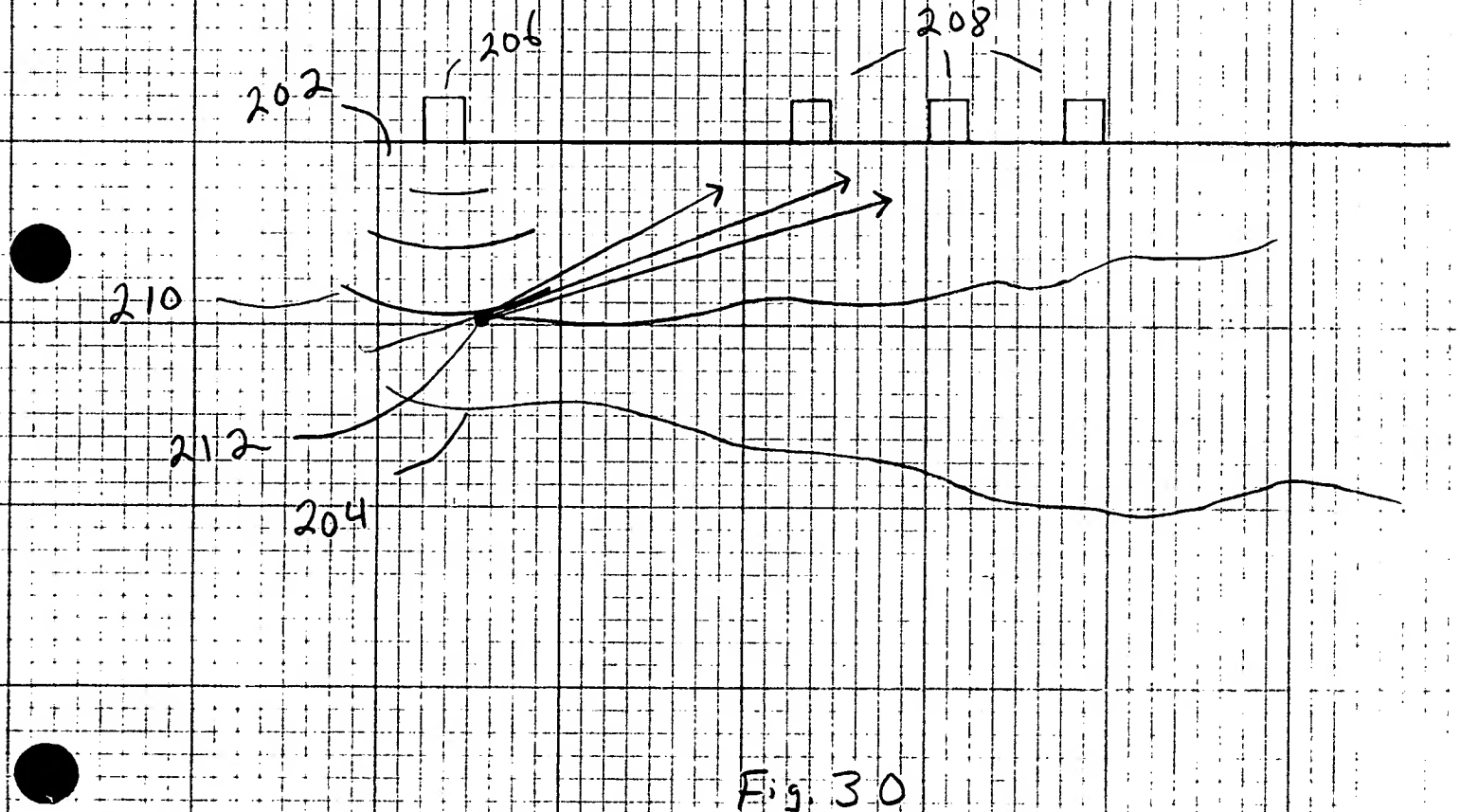
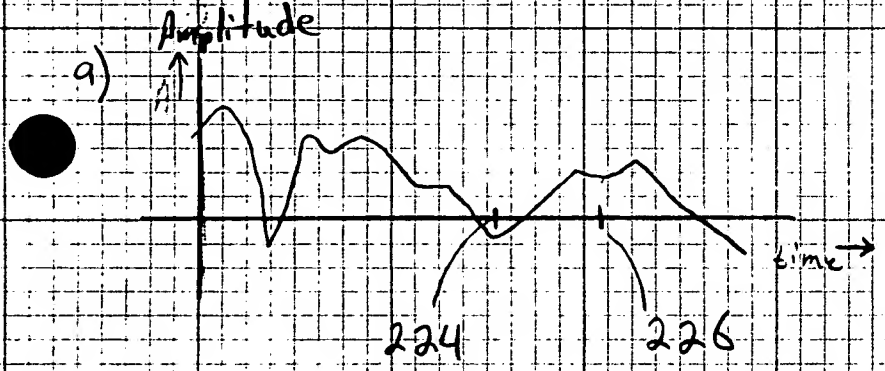
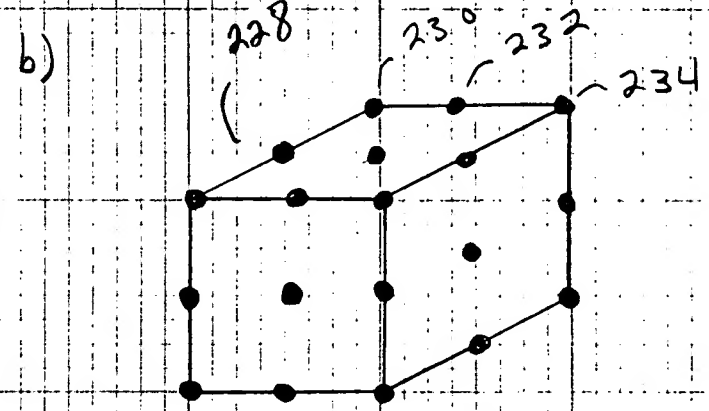


Fig. 3 0





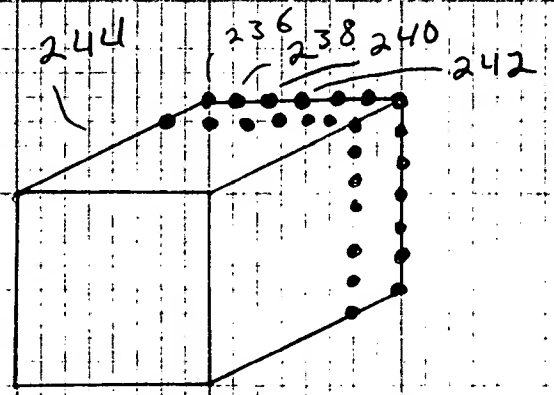


Fig. 31

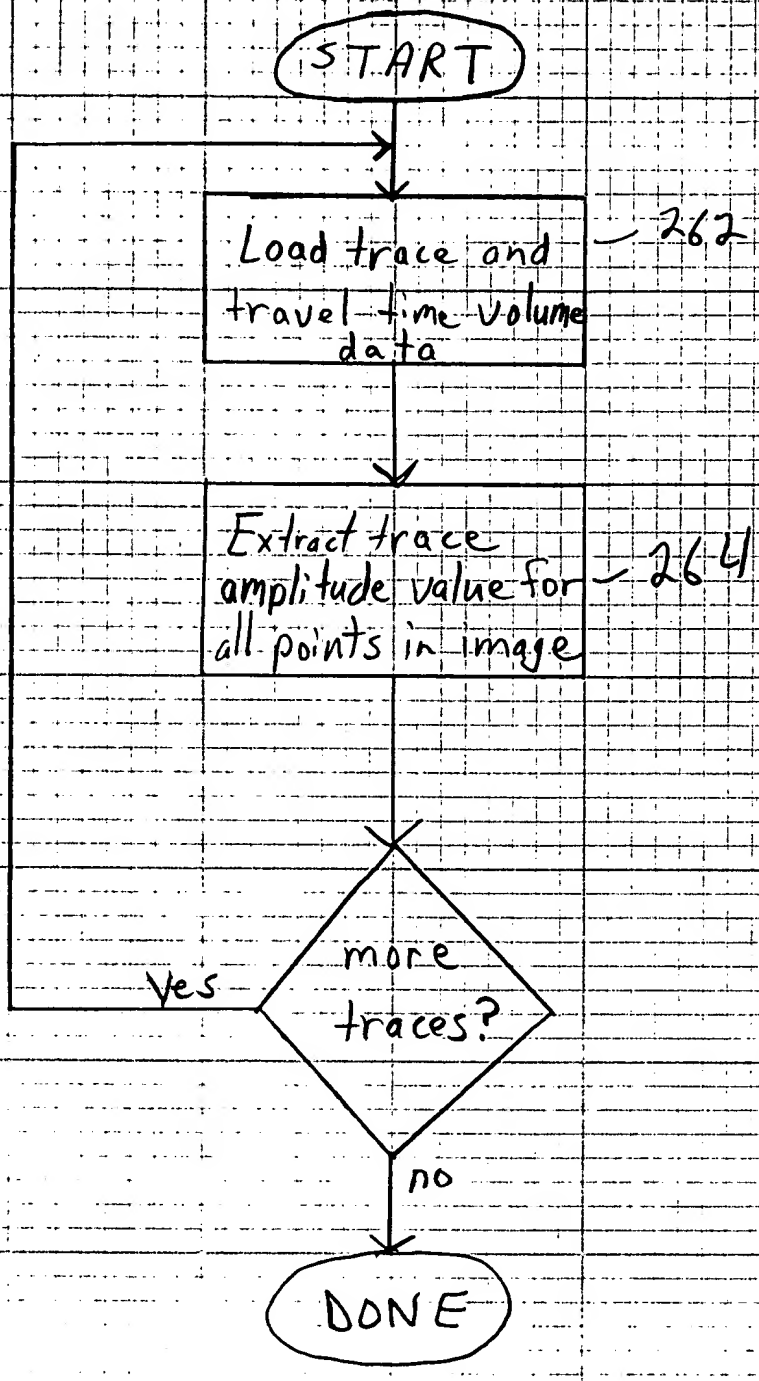

START
Load trace and travel-time volume data
262
Extract trace amplitude value for all points in image
264
more traces?
yes
no
DONE

Fig. 32

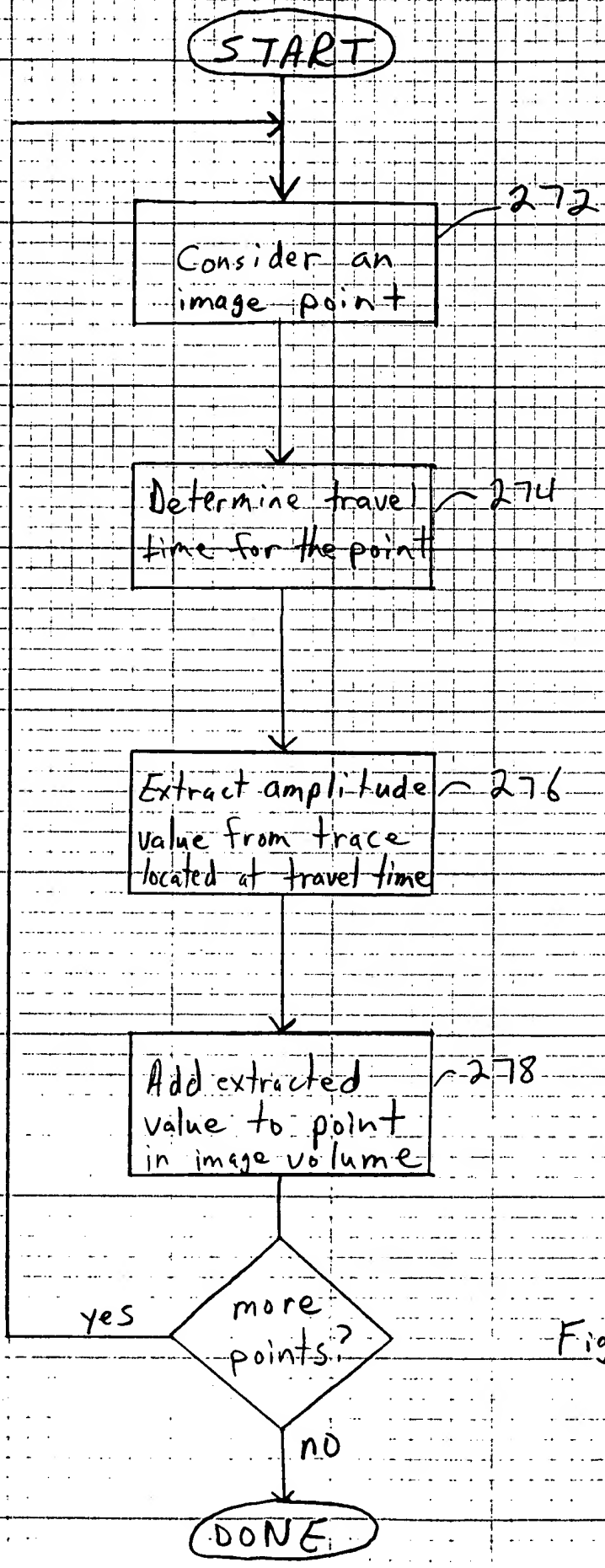

START
Consider an image point
272
Determine travel time for the point
274
Extract amplitude value from trace located at travel time
276
Add extracted value to point in image volume
278
more points?
yes
no
DONE

Fig. 33

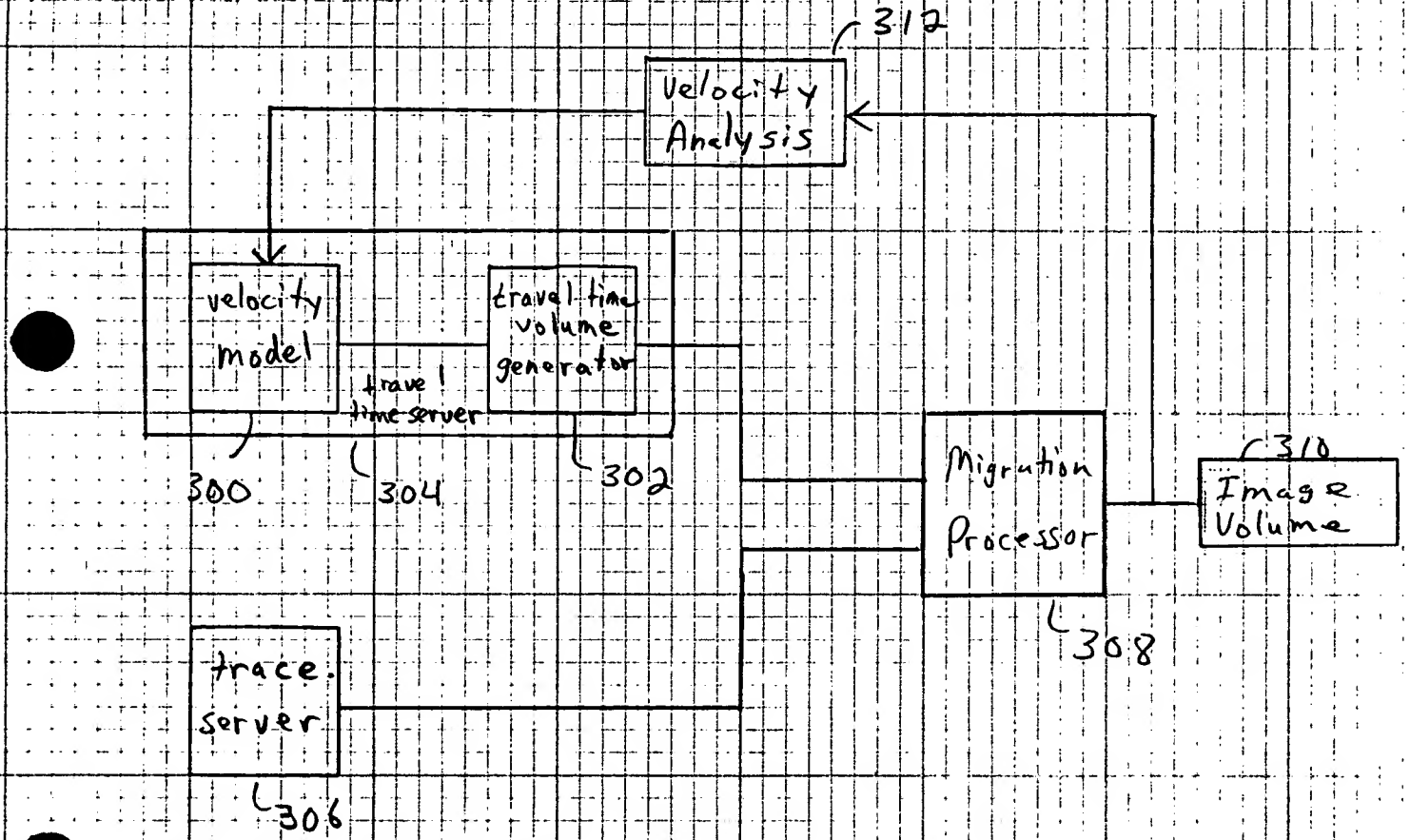


Fig. 34

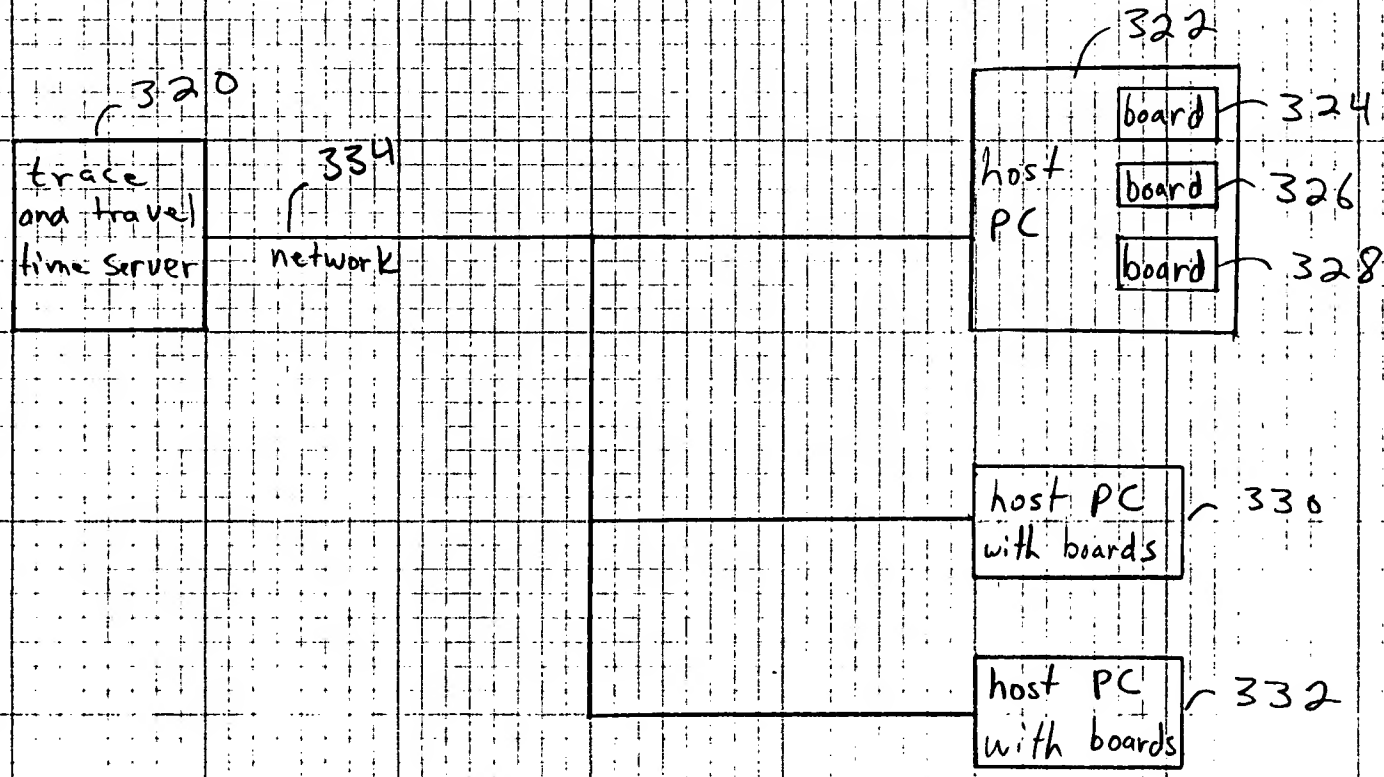
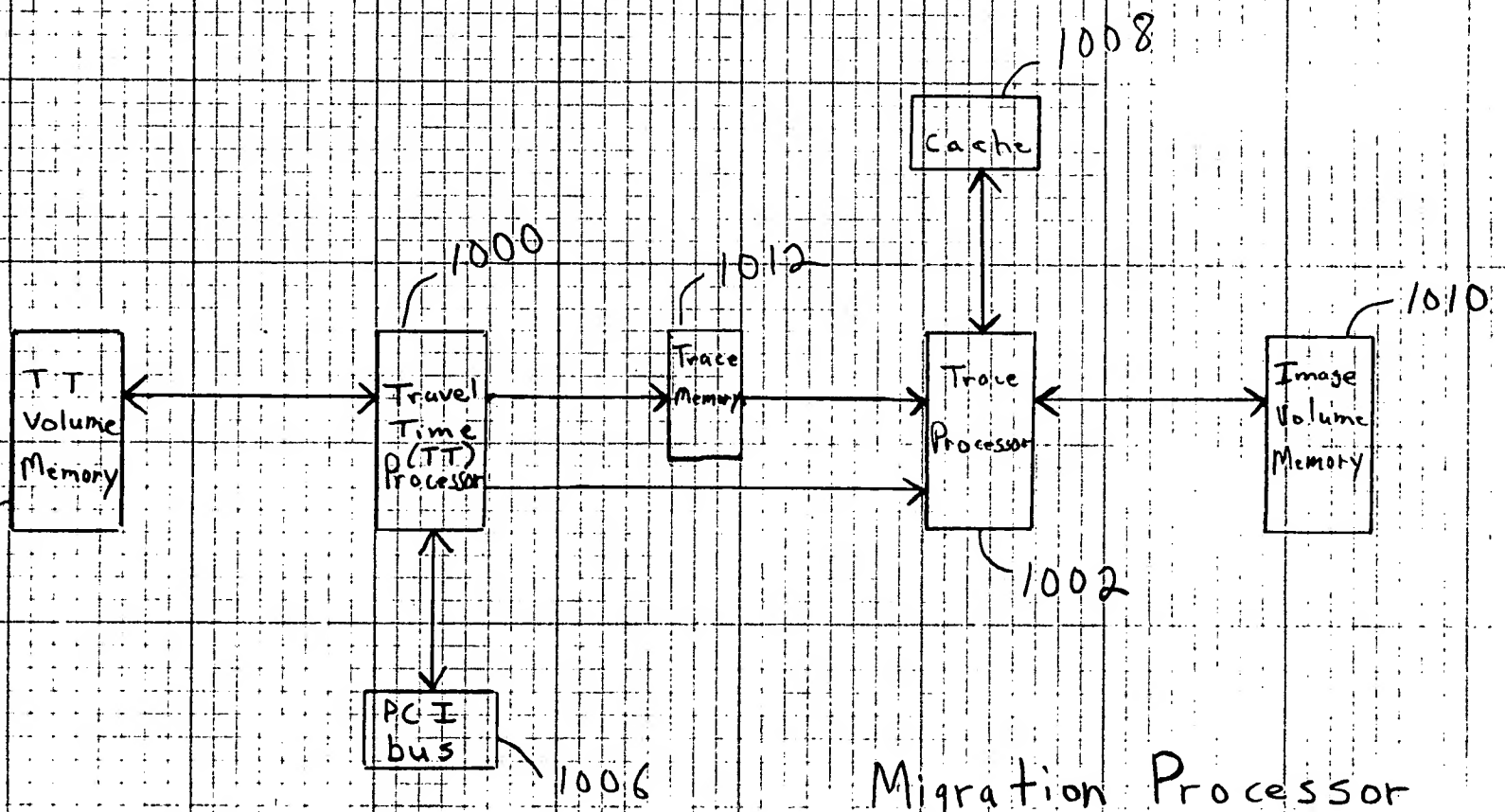


Fig. 35



Migration Processor
Top Level

Fig. 36

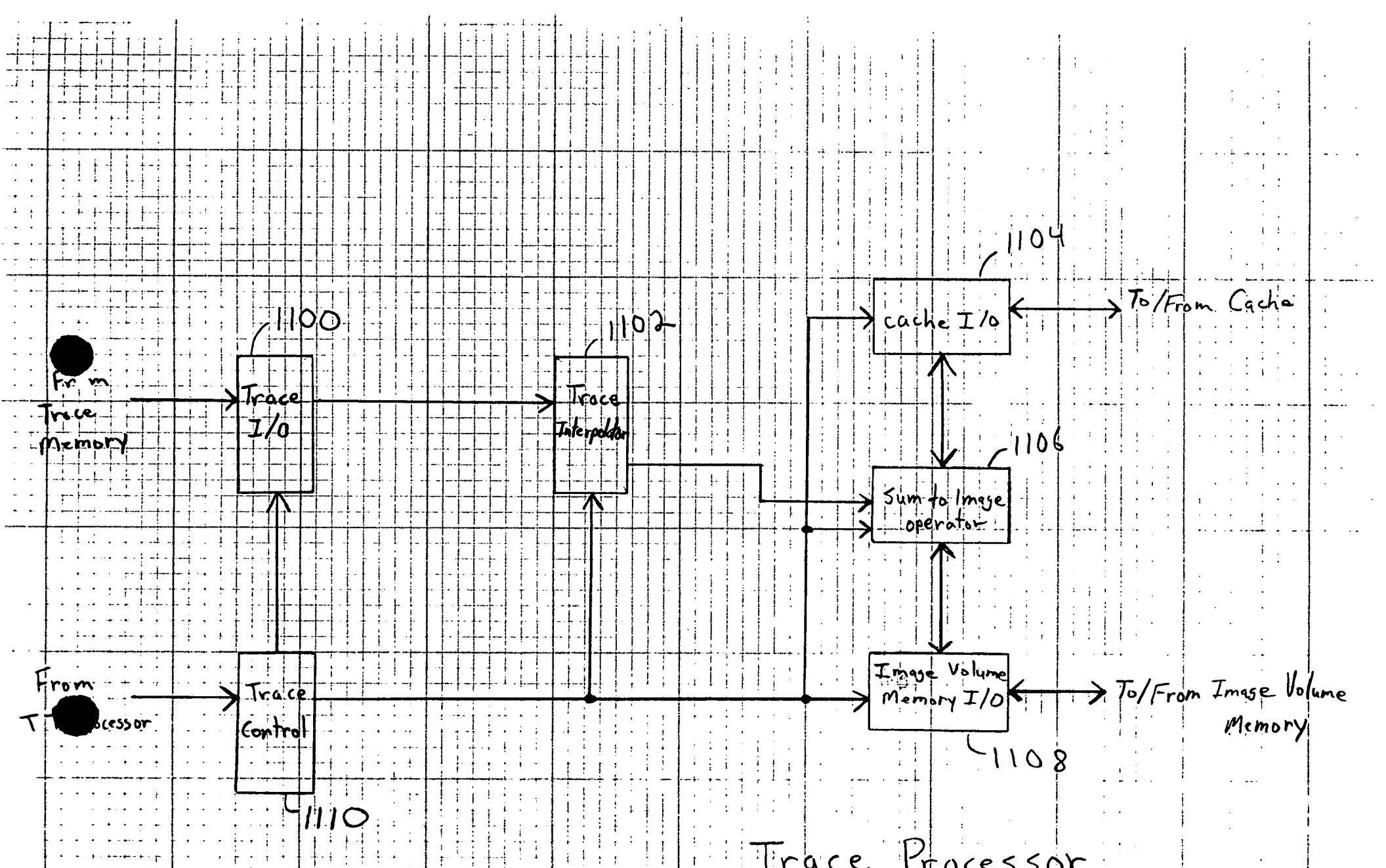
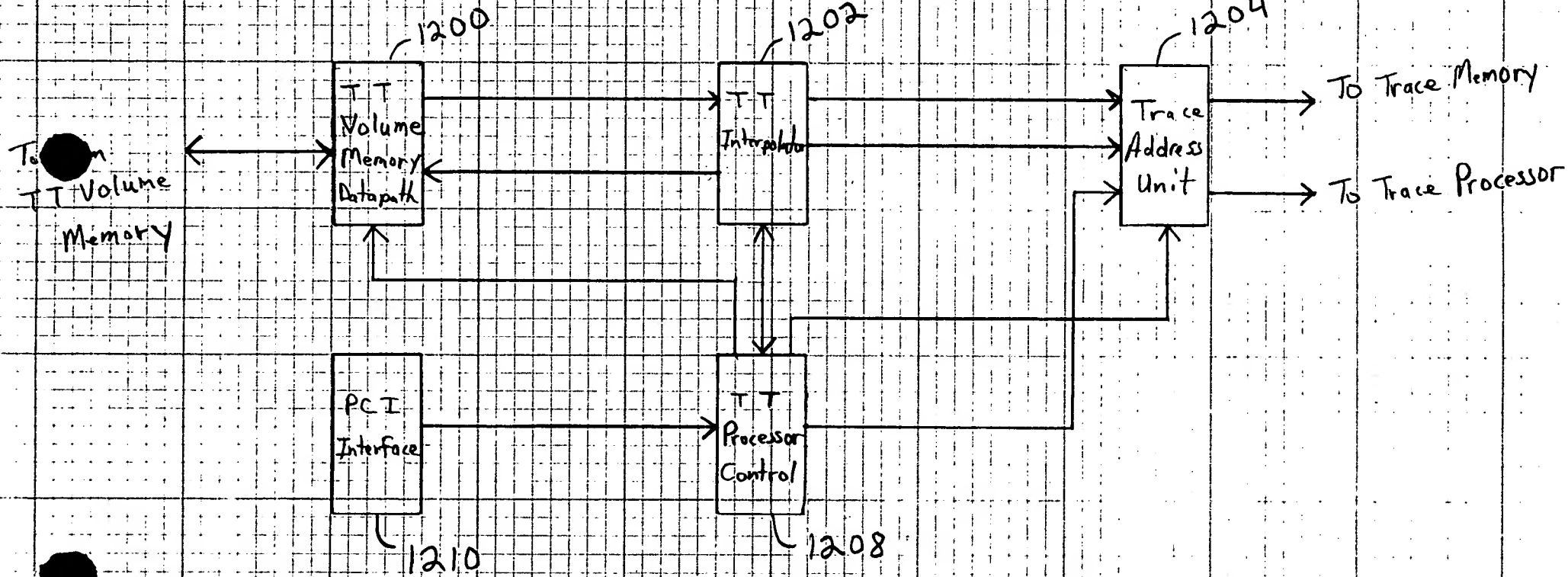
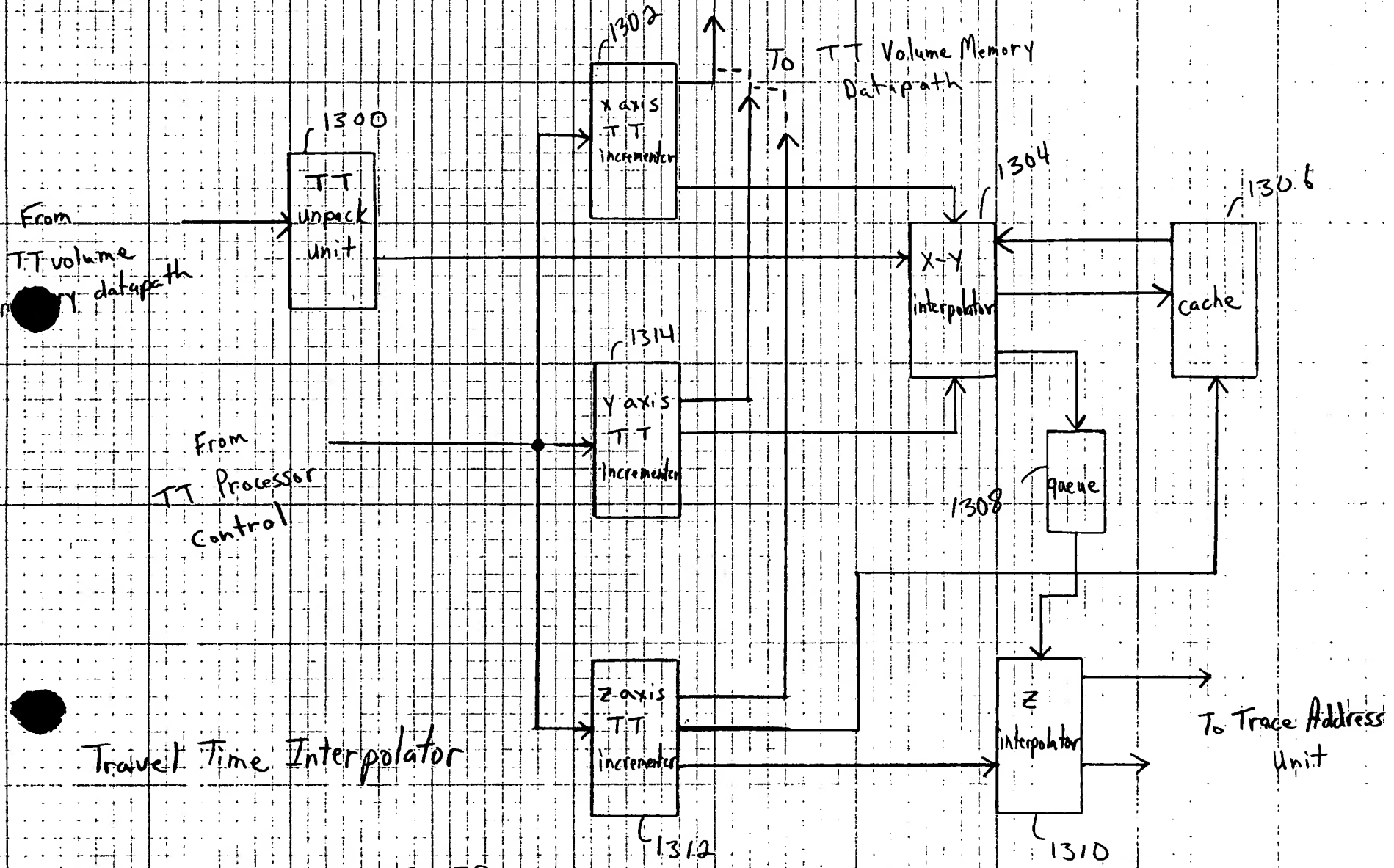


Fig. 37



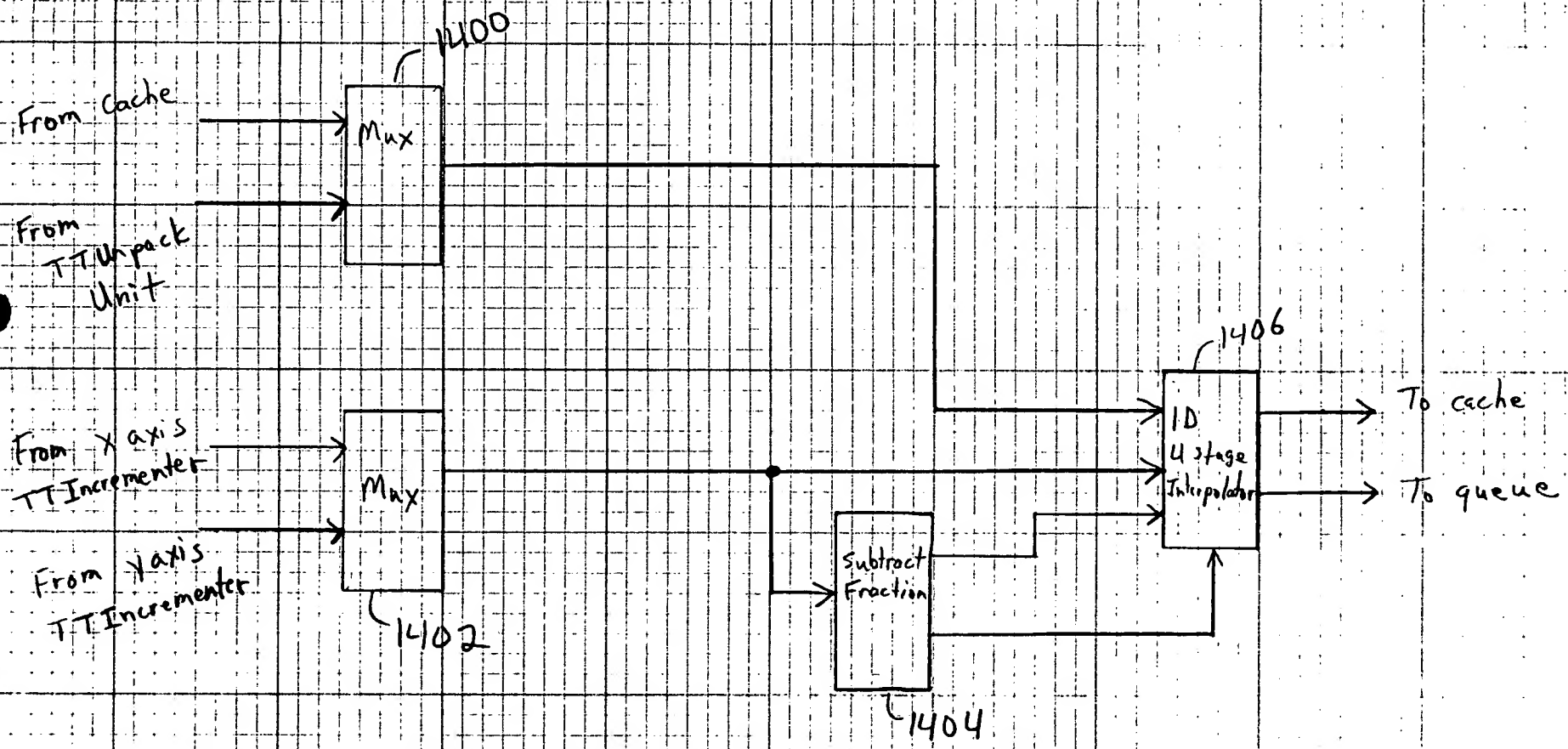
Travel Time Processor

Fig. 38



Travel Time Interpolator

Fig. 39



X-Y Interpolator

Fig. 40

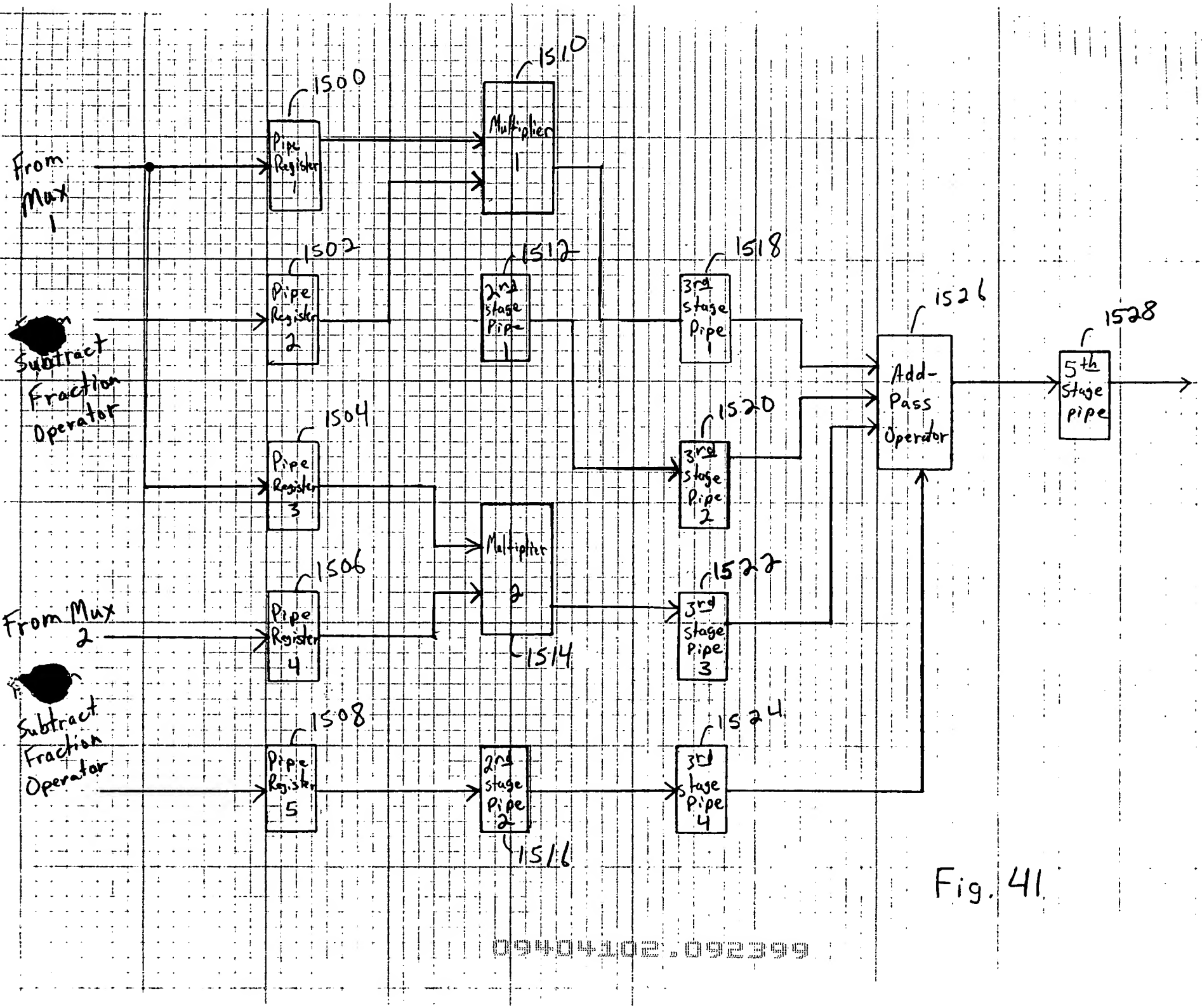


Fig. 41